

A Total Variation and Group Sparsity Based Tensor Optimization Model for Video Rain Streak Removal

Ye-Tao Wang, Xi-Le Zhao*, Tai-Xiang Jiang*, Liang-Jian Deng,
Tian-Hui Ma, Yue-Tian Zhang, Ting-Zhu Huang

*School of Mathematical Sciences, University of Electronic Science and Technology of China,
Chengdu, Sichuan, 611731, P. R. China*

Abstract

Rain streak removal is an important issue of the outdoor vision system and has been investigated extensively. In this paper, we propose a novel tensor optimization model for video rain streak removal by fully considering the discriminatively intrinsic characteristics of rain streaks and clean videos. In specific, rain streaks are group sparse and smooth along the rain streaks' direction; the clean videos are smooth along the perpendicular direction of rain streaks and the time direction. For rain streaks, we use the $l_{2,1}$ norm to enhance the group sparsity and the *Unidirectional Total Variation* (UTV) to promote the smoothness along rain streaks' direction. For clean videos, we use two UTV to enhance the smoothness along the perpendicular direction of rain streaks and the time direction. We develop an efficient *alternating direction method of multipliers* (ADMM) algorithm to solve the proposed model. Experiments on synthetic and real data demonstrate the superiority of the proposed method over state-of-the-art methods in terms of both quantitative and qualitative assessments.

Keywords: video rain streak removal, group sparsity, unidirectional total variation, tensor optimization model, alternating direction method of multipliers.

1. INTRODUCTION

Bad weather impairs visibility of an image and introduces undesirable interference that can severely hinder the follow-up processing (e.g., object detection,

*Corresponding author

Email address: xlzhao122003@163.com (Xi-Le Zhao)

recognition, and tracking [1, 2, 3, 4, 5]). This paper mainly focuses on the rain streak removal problem [6, 7, 8, 9, 10, 11].

The degradation of rainy images is generally modeled as the sum of the unknown clean images and the rain streaks. A single rainy image is generally modeled as $\mathbf{O} = \mathbf{B} + \mathbf{R}$ [7, 12, 13], where $\mathbf{O} \in \mathbb{R}^{m \times n}$, $\mathbf{B} \in \mathbb{R}^{m \times n}$, and $\mathbf{R} \in \mathbb{R}^{m \times n}$ are the observed rainy image, the unknown clean image, and the rain streaks, respectively. This model can be extended to the video case: $\mathcal{O} = \mathcal{B} + \mathcal{R}$, where \mathcal{O} , \mathcal{B} , and $\mathcal{R} \in \mathbb{R}^{m \times n \times t}$ are the observed rainy video, the unknown clean video, and the rain streaks, respectively. The goal of rain streak removal is to estimate the clean images from its rainy version. This typical inverse problem is often solved by regularization methods which are based on additional prior knowledge.

Existing rain streak removal algorithms can be categorized into two classes: the single image rain streak removal algorithms and the video rain streak removal algorithms. For the single image rain streak removal, Kang et al. [7] decomposed a rainy image into low-frequency (LF) and high-frequency (HF) components using a bilateral filter and then performed *morphological component analysis* (MCA)-based dictionary learning and sparse coding to separate the rain streaks in the HF component. However, learning HF image bases typically results in a loss of detailed image information. To alleviate this problem, Sun et al. [14] exploited the structural similarity of the derived HF image bases. Nevertheless, the backgrounds estimated using their method still tend to be blurry. Chen et al. [12] considered the pattern of the rain streaks and the smoothness of the background, but the constraints in their objective function were not sufficiently strong. Discriminative sparse coding was adopted by Luo et al. [8]. Their method preserves the clean content well but is not able to remove most of the rain streaks. The recent work by Li et al. [13] was the first to utilize *Gaussian mixture model* (GMM) patch priors for rain streak removal, with the ability to account for rain streaks of different orientations and scales. Nonetheless, their method tends to yield over-smooth clean images; i.e., the details of the clean image content are not preserved well. To cope with this issue, Zhu et al. [15] proposed a joint bi-layer optimization method progressively separate rain streaks from background details, in which the gradient statistics are analyzed. In [16], the directional property of rain streaks received attentions. The recently developed deep learning technique is also applied to the single image rain streak removal task [17, 18].

For the video rain streak removal, Garg et al. [19] firstly raised a video rain streak removal method with comprehensive analysis of the visual effects of rain streaks on an imaging system. Since then, multiple methods have been proposed for the video rain streak removal and attained good rain removing performance in

videos with different rain circumstances. Tripathi et al. [20] took the spatiotemporal properties into consideration. In [12], the similarity and repeatability of rain streaks were utilized, and a generalized low-rank appearance model was proposed. Additionally, comprehensive early existing video-based methods are reviewed in [21]. Kim et al. [6] considered the temporal correlation of rain streaks and the low-rank nature of clean videos, but the effectiveness of their method is still low for certain dynamic videos recorded by dynamic cameras. Very recently, the rain streaks were stochastically modeled as a mixture of Gaussians in [22]. In [23], a novel tensor-based video rain streak removal approach was proposed, with considering numerous discriminative prior information.

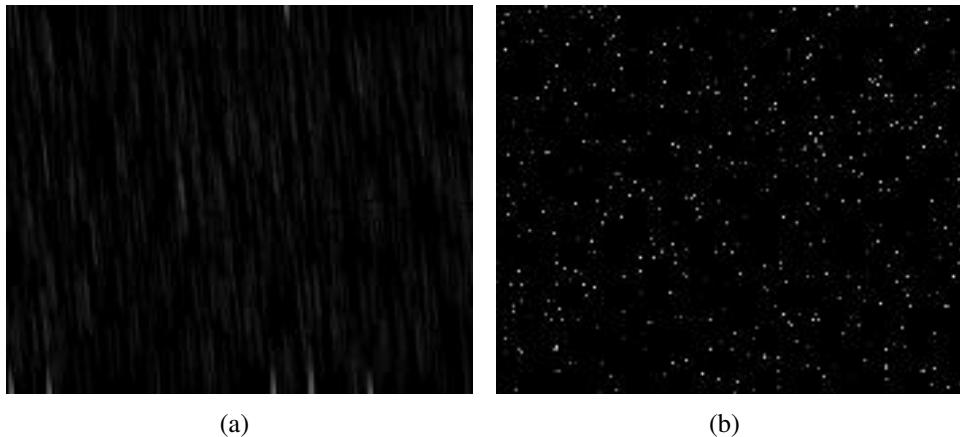


Figure 1: (a) The rain streaks, (b) A random sparse image.

In [23], Jiang et at. proposed the model as

$$\begin{aligned} \min_{\mathcal{B}, \mathcal{R}} \quad & \alpha_1 \|\nabla_x \mathcal{R}\|_1 + \alpha_2 \|\mathcal{R}\|_1 + \alpha_3 \|\nabla_y \mathcal{B}\|_1 + \alpha_4 \|\nabla_t \mathcal{B}\|_1 + \alpha_5 \|\mathcal{B}\|_* , \\ \text{s.t.} \quad & \mathcal{O} = \mathcal{B} + \mathcal{R}, \quad \mathcal{B}, \mathcal{R} \geq 0, \end{aligned} \quad (1)$$

where ∇_x , ∇_y , and ∇_t are the derivative operators along rain streaks direction, the perpendicular direction of rain streaks, and time direction, respectively. For simplicity, we assume that the rain streaks direction and the perpendicular direction of rain streaks are the vertical direction and the horizontal direction, respectively.

However, model 1 has two drawbacks. First, the rain streaks are not only sparse and but also group sparse; see Figure 1. Second, the clean video does not exhibit obvious low-rankness; see Figure 2. Hence, there is room for improvement. Based on the above observations, we introduce the group sparsity regularizer

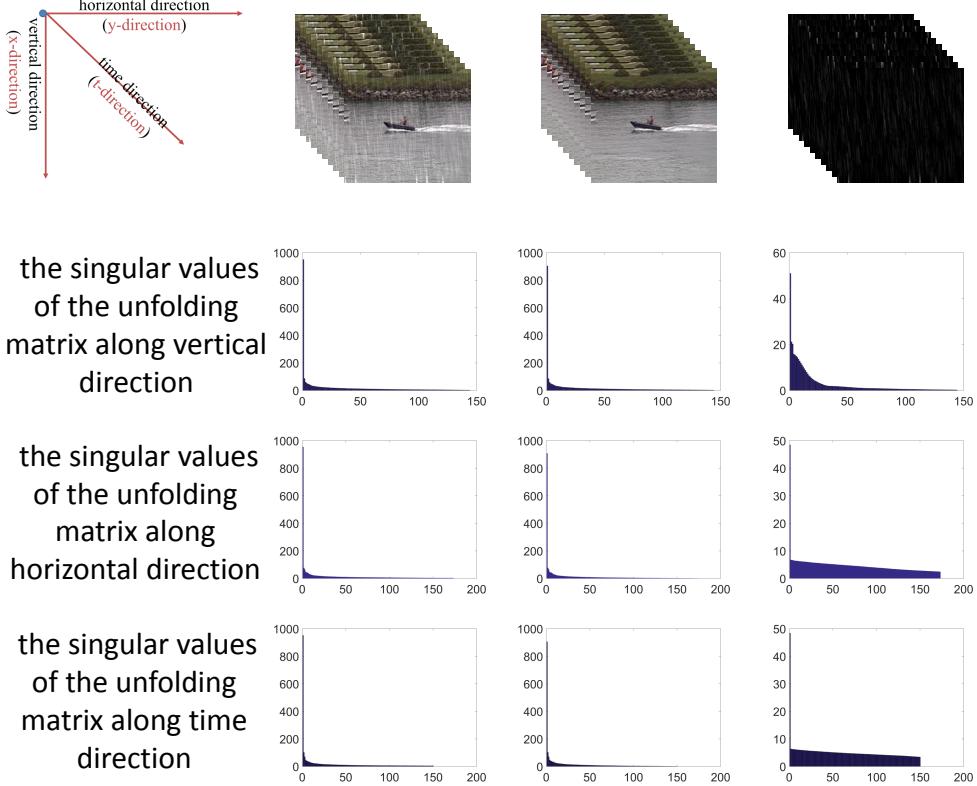


Figure 2: From left to right: the singular values of unfolding matrices of the rainy video, the clean video, and the rain streaks.

for rain streaks and disuse the low-rankness regularizer for the clean video. The novel tensor optimization model consists of the group sparsity regularizer and the *Unidirectional Total Variation* (UTV) regularizer along vertical direction for rain streaks and the UTV regularizers along horizontal direction and time direction for clean videos. We build model as

$$\begin{aligned} \arg \min_{\mathcal{B}, \mathcal{R}} \quad & \alpha_1 \|\mathcal{R}\|_{2,1} + \alpha_2 \|\nabla_x \mathcal{R}\|_1 + \alpha_3 \|\nabla_y \mathcal{B}\|_1 + \alpha_4 \|\nabla_t \mathcal{B}\|_1, \\ \text{s.t.} \quad & \mathcal{O} = \mathcal{B} + \mathcal{R}, \quad \mathcal{B}, \mathcal{R} \geq 0. \end{aligned} \quad (2)$$

To solve the proposed model, we develop an efficient ADMM [24, 25, 26, 27] algorithm. Experimental results demonstrate the superior of the proposed method qualitatively and visually.

The paper is organized as follows. In Sec. 2, some notations and the basic knowledge are introduced. In Sec. 3, the proposed model and proposed algorithm are presented. Experimental results are reported in Sec. 4. Finally, we draw some conclusions in Sec. 5.

2. TENSOR BASICS

Following [23, 28, 29], we use lower case letters (e.g., x) for scalars, bold lower case letters (e.g., \mathbf{x}) for vectors, bold upper case letters (e.g., \mathbf{X}) for matrixes, and bold upper calligraphic letters (e.g., \mathcal{X}) for tensors. An n -mode tensor is denoted as $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_n}$. Its elements are denoted as x_{i_1, \dots, i_n} , where $1 \leq i_k \leq I_k$ and $1 \leq k \leq n$. The inner product of two same-size tensors is defined as

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1, i_2, \dots, i_n} x_{i_1, i_2, \dots, i_n} \times y_{i_1, i_2, \dots, i_n}. \quad (3)$$

Based on (3), the Frobenius norm of a tensor is defined as

$$\|\mathcal{X}\|_F := \langle \mathcal{X}, \mathcal{X} \rangle^{\frac{1}{2}} = \left(\sum_{i_1, i_2, \dots, i_n} |x_{i_1, i_2, \dots, i_n}|^2 \right)^{\frac{1}{2}}. \quad (4)$$

For an n -mode tensor, we define the derivative along the k -th direction of \mathcal{X} as $\nabla_k \mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_n}$ in the cyclic boundary condition, where the elements of $\nabla_k \mathcal{X}$ obey that

$$(\nabla_k \mathcal{X})_{i_1, i_2, \dots, i_k, \dots, i_n} = x_{i_1, i_2, \dots, i_k, \dots, i_n} - x_{i_1, i_2, \dots, (i_k-1), \dots, i_n}.$$

When $i_k = 1$, the $i_k - 1$ will be I_k . The “unfold” operation along the k -th direction on a tensor \mathcal{X} is defined as

$$\text{unfold}_k(\mathcal{X}) = \mathbf{X}_{(k)} \in \mathbb{R}^{I_k \times (I_1 \dots I_{k-1} I_{k+1} \dots I_n)}. \quad (5)$$

The projection operator “fold” is defined as

$$\text{fold}_k(\mathbf{X}_{(k)}) = \mathcal{X}. \quad (6)$$

Based on the unfolding rule (5) and folding rule (6), the tensor and the matrix can be transformed to each other. It is easy to obtain that, for any $1 \leq k \leq n$,

$$\|\mathcal{X}\|_F = \|\mathbf{X}_{(k)}\|_F, \quad \langle \mathcal{X}, \mathcal{Y} \rangle = \langle \mathbf{X}_{(k)}, \mathbf{Y}_{(k)} \rangle,$$

and

$$\nabla_k \mathcal{X} = \text{fold}_k(\nabla_1 \text{unfold}_k(\mathcal{X})).$$

Suppose $\mathbf{x} \in \mathbb{R}^n$ is a group sparse vector. Let $\{\mathbf{x}_{g_i} \in \mathbb{R}^{n_i} : i = 1, \dots, s\}$ be the grouping of \mathbf{x} , where $g_i \subseteq \{1, 2, \dots, n\}$ is an index set corresponding to the i -th group, and \mathbf{x}_{g_i} denotes the subvector of \mathbf{x} indexed by g_i [30]. Generally, g_i 's can be any index sets, and they are predefined based on prior knowledge. The $l_{2,1}$ norm is defined as follows:

$$\|\mathbf{x}\|_{2,1} = \sum_{i=1}^s \|\mathbf{x}_{g_i}\|_2.$$

$l_{2,1}$ norm is known to facilitate group sparsity [30]. For the matrix, each column is considered as a group. Thus $l_{2,1}$ norm for a matrix is usually denoted as

$$\|\mathbf{X}\|_{2,1} = \sum_{i=1}^s \|\mathbf{x}_{g_i}\|_2.$$

Here, g_i 's are the column index set. Since one column is treated as a group, we can extend $l_{2,1}$ norm from the matrix to the tensor as

$$\|\mathcal{X}\|_{2,1} = \|\text{unfold}_1(\mathcal{X})\|_{2,1}.$$

More extensive overview of group sparsity can be found in [30].

3. THE PROPOSED METHOD

This section gives the proposed model and the algorithm for rain streak removal.

3.1. Proposed model

Without loss of generality, we use \mathcal{O} , \mathcal{B} , and \mathcal{R} to represent the rainy video, the target clean video, and the rain streaks, respectively. We recall the proposed model:

$$\begin{aligned} & \arg \min_{\mathcal{B}, \mathcal{R}} \alpha_1 \|\mathcal{R}\|_{2,1} + \alpha_2 \|\nabla_x \mathcal{R}\|_1 + \alpha_3 \|\nabla_y \mathcal{B}\|_1 + \alpha_4 \|\nabla_t \mathcal{B}\|_1, \\ & \text{s.t. } \mathcal{O} = \mathcal{B} + \mathcal{R}, \quad \mathcal{B}, \mathcal{R} \geq 0, \end{aligned} \tag{7}$$

where ∇_x , ∇_y , and ∇_t are the derivative operators along the vertical direction, the horizontal direction, and the time direction, respectively. In what followings, we will explain all components in our model in details.

Group sparsity of the rain streaks: The rain component is sparser than the clean video, and the rain component exhibits line pattern structure rather than being randomly distributed just like Figure 1. Therefore, we use the term $\|\mathcal{R}\|_{2,1}$ to characterize the group sparse which can simultaneously enhance the sparsity and preserve the line pattern. It is superior over the sparsity itself used in [23].

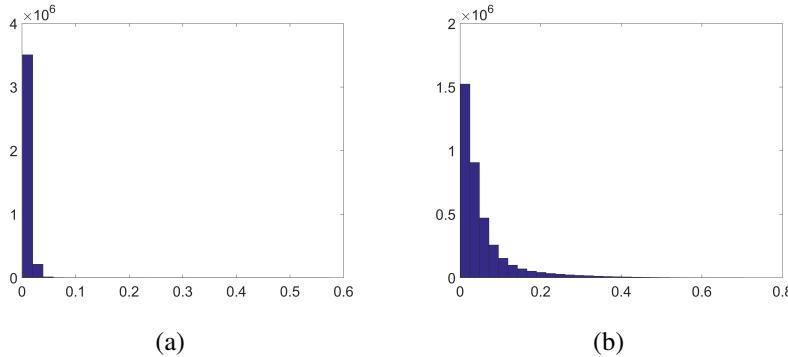


Figure 3: (a) The histogram of the absolute values of the derivatives along the vertical direction of the rain streaks. (b) The histogram of the absolute values of the derivatives along the vertical direction of the clean video.

The smoothness along the rain streak direction of the rain streaks: The rain streaks share similar directions. When the angle between the direction of rain streaks and the vertical direction is small, the derivatives of rain streaks and the clean video along the vertical direction are different, i.e., the derivatives along the vertical direction of rain streaks are more sparse as compared with those of the clean video; see Figure 3. Therefore, we use the l_1 norm of $\nabla_x \mathcal{R}$ to enhance the smoothness along the vertical direction of the rain streaks.

The smoothness along the horizontal direction of the clean video: Natural images are piecewise smooth, which indicates that the derivatives of frames in a video are not dense along vertical and horizontal directions. The vertical rain streaks destroy the smoothness along the horizontal direction. Compared with the rain streaks, the derivatives of the clean video are sparse along the horizontal direction. As a result, the derivatives along the horizontal direction of rain streaks are dense, which is shown in Figure 4. Therefore, we use the l_1 norm of $\nabla_y \mathcal{B}$ to enhance the smoothness along the horizontal direction of the clean video.

The smoothness along the time direction of the clean video: Since that a video maintains at least 25 frames per second, there is a strong smoothness along time direction. The derivatives of the clean video are sparse along the time

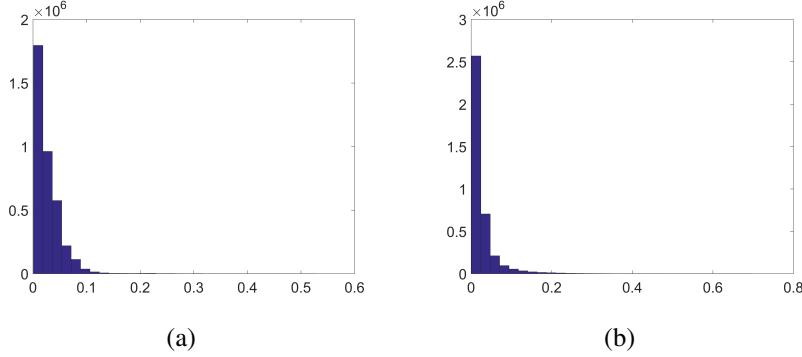


Figure 4: (a) The histogram of the absolute values of the derivatives along the horizontal direction of the rain streaks. (b) The histogram of the absolute values of the derivatives along the horizontal direction of the clean video.

direction. However, the rain streaks are not smooth. Because of its high velocity, its smoothness is broken. As displayed in Figure 5, the derivatives along the time direction of the clean video are sparse while those of the rain streaks. Therefore, we use the l_1 norm of $\nabla_t \mathcal{B}$ to enhance the smoothness along the time direction of the clean video.

Discussion of low-rankness: Meanwhile, we discard the low-rankness regularizer which is considered in [23]. The clean video is low-rank only when it is static, but not the case even if there is only a light object moving in the clean video. Usually the low-rankness regularizer will be slacked to the singular values of three unfolding matrixes of the video in quantitative analysis. From the *singular value decomposition* (SVD) [31] of rain streaks and clean video in Figure 2, it can be found the singular value of clean video does not have zero elements in any directions, and the singular values of rain streaks are smaller than those of clean video.

3.2. Proposed algorithm

The proposed model (7) is a convex optimization problem which can be solved by various of convex optimization algorithms. We adopt the ADMM, an effective strategy for solving large scale optimization problems, to solve it. After introducing four auxiliary tensors \mathcal{Y} , \mathcal{S} , \mathcal{X} , and $\mathcal{T} \in \mathbb{R}^{m \times n \times t}$, we rewrite the proposed

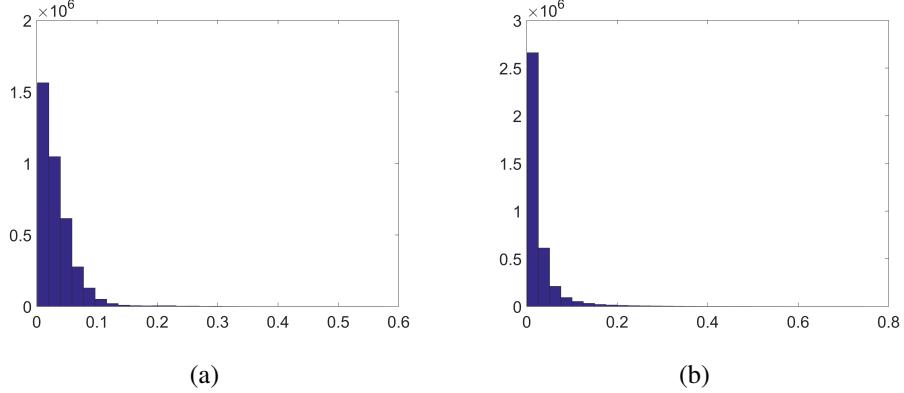


Figure 5: (a) The histogram of the absolute values of the derivatives along the time direction of the rain streaks. (b) The histogram of the absolute values of the derivatives along the time direction of the clean video.

model (7) as the following equivalent constrained problem:

$$\begin{aligned}
 & \arg \min_{\mathcal{R}, \mathcal{Y}, \mathcal{S}, \mathcal{X}, \mathcal{T}} \quad \alpha_1 \|\mathcal{Y}\|_{2,1} + \alpha_2 \|\mathcal{S}\|_1 + \alpha_3 \|\mathcal{X}\|_1 + \alpha_4 \|\mathcal{T}\|_1, \\
 \text{s.t. } & \mathcal{Y} = \mathcal{R}, \\
 & \mathcal{S} = \nabla_x \mathcal{R}, \\
 & \mathcal{X} = \nabla_y (\mathcal{O} - \mathcal{R}), \\
 & \mathcal{T} = \nabla_t (\mathcal{O} - \mathcal{R}), \\
 & \mathcal{O} \geq \mathcal{R} \geq 0.
 \end{aligned} \tag{8}$$

Then the augmented Lagrangian function of (8) is:

$$\begin{aligned}
 L_\beta(\mathcal{R}, \mathcal{Y}, \mathcal{S}, \mathcal{X}, \mathcal{T}, \boldsymbol{\Lambda}) = & \alpha_1 \|\mathcal{Y}\|_{2,1} + \alpha_2 \|\mathcal{S}\|_1 + \alpha_3 \|\mathcal{X}\|_1 + \alpha_4 \|\mathcal{T}\|_1 \\
 & + \langle \boldsymbol{\Lambda}_1, \mathcal{Y} - \mathcal{R} \rangle + \frac{\beta_1}{2} \|\mathcal{Y} - \mathcal{R}\|_F^2 \\
 & + \langle \boldsymbol{\Lambda}_2, \mathcal{S} - \nabla_x \mathcal{R} \rangle + \frac{\beta_2}{2} \|\mathcal{S} - \nabla_x \mathcal{R}\|_F^2 \\
 & + \langle \boldsymbol{\Lambda}_3, \mathcal{X} - \nabla_y (\mathcal{O} - \mathcal{R}) \rangle + \frac{\beta_3}{2} \|\mathcal{X} - \nabla_y (\mathcal{O} - \mathcal{R})\|_F^2 \\
 & + \langle \boldsymbol{\Lambda}_4, \mathcal{T} - \nabla_t (\mathcal{O} - \mathcal{R}) \rangle + \frac{\beta_4}{2} \|\mathcal{T} - \nabla_t (\mathcal{O} - \mathcal{R})\|_F^2,
 \end{aligned} \tag{9}$$

where $\boldsymbol{\Lambda} = [\boldsymbol{\Lambda}_1, \boldsymbol{\Lambda}_2, \boldsymbol{\Lambda}_3, \boldsymbol{\Lambda}_4]$ are Lagrange multipliers and $\beta = [\beta_1, \beta_2, \beta_3, \beta_4]$ are four positive penalty parameters. This joint minimization problem can be decomposed into five subproblems which can be easily solved. By separating the variables of (9) into two groups: \mathcal{R} and $(\mathcal{Y}, \mathcal{S}, \mathcal{X}, \mathcal{T})$, (9) fits the framework of ADMM. It requests us to solve variables of each group by keeping another group fixed. The solution of the five subproblems will be introduced in the following.

\mathcal{Y} sub-problem: With other variables fixed, the \mathcal{Y} sub-problem is

$$\arg \min_{\mathcal{Y}} \alpha_1 \|\mathcal{Y}\|_{2,1} + \frac{\beta_1}{2} \|\mathcal{Y} - \mathcal{R} + \frac{\boldsymbol{\Lambda}_1}{\beta_1}\|_F^2, \quad (10)$$

which has a closed-form solution by the soft-shrinkage formula [30], thus \mathcal{Y} could be updated as

$$\mathcal{Y}_{g_i}^{t+1} = \max \left(\|\mathcal{Q}_{g_i}\|_2 - \frac{\alpha_1}{\beta_1}, 0 \right) \frac{\mathcal{Q}_{g_i}}{\|\mathcal{Q}_{g_i}\|_2}, \quad (11)$$

where \mathcal{Q}_{g_i} denotes the i -th group of the video.

\mathcal{S} , \mathcal{X} , and \mathcal{T} sub-problems: With other variables fixed, \mathcal{S} , \mathcal{X} , and \mathcal{T} subproblems are

$$\begin{aligned} \arg \min_{\mathcal{S}} \alpha_2 \|\mathcal{S}\|_1 + \frac{\beta_2}{2} \|\mathcal{S} - \nabla_x \mathcal{R} + \frac{\boldsymbol{\Lambda}_2}{\beta_2}\|_F^2 \\ \arg \min_{\mathcal{X}} \alpha_3 \|\mathcal{X}\|_1 + \frac{\beta_3}{2} \|\mathcal{X} - \nabla_y (\mathcal{O} - \mathcal{R}) + \frac{\boldsymbol{\Lambda}_3}{\beta_3}\|_F^2 \\ \arg \min_{\mathcal{T}} \alpha_4 \|\mathcal{T}\|_1 + \frac{\beta_4}{2} \|\mathcal{T} - \nabla_t (\mathcal{O} - \mathcal{R}) + \frac{\boldsymbol{\Lambda}_4}{\beta_4}\|_F^2, \end{aligned} \quad (12)$$

which have closed-form solutions by soft-thresholding , thus \mathcal{S} , \mathcal{X} , and \mathcal{T} could be updated as

$$\mathcal{S}^{(t+1)} = \text{Shrink}_{\frac{\alpha_2}{\beta_2}} \left(\nabla_x \mathcal{R}^{(t)} - \frac{\boldsymbol{\Lambda}_2^{(t)}}{\beta_2} \right), \quad (13)$$

$$\mathcal{X}^{(t+1)} = \text{Shrink}_{\frac{\alpha_3}{\beta_3}} \left(\nabla_y (\mathcal{O} - \mathcal{R}^{(t)}) - \frac{\boldsymbol{\Lambda}_3^{(t)}}{\beta_3} \right), \quad (14)$$

$$\mathcal{T}^{(t+1)} = \text{Shrink}_{\frac{\alpha_4}{\beta_4}} \left(\nabla_t (\mathcal{O} - \mathcal{R}^{(t)}) - \frac{\boldsymbol{\Lambda}_4^{(t)}}{\beta_4} \right). \quad (15)$$

R-subproblem: The \mathcal{R} sub-problem is a least squares problem:

$$\begin{aligned} \arg \min_{\mathcal{R}} \quad & \frac{\beta_1}{2} \|\mathcal{Y} - \mathcal{R} + \frac{\boldsymbol{\Lambda}_1}{\beta_1}\|_F^2 + \frac{\beta_2}{2} \|\mathcal{S} - \nabla_x \mathcal{R} + \frac{\boldsymbol{\Lambda}_2}{\beta_2}\|_F^2 \\ & + \frac{\beta_3}{2} \|\mathcal{X} - \nabla_y (\mathcal{O} - \mathcal{R}) + \frac{\boldsymbol{\Lambda}_3}{\beta_3}\|_F^2 + \frac{\beta_4}{2} \|\mathcal{T} - \nabla_t (\mathcal{O} - \mathcal{R}) + \frac{\boldsymbol{\Lambda}_4}{\beta_4}\|_F^2. \end{aligned}$$

With the problem is transformed to

$$\begin{aligned} (\beta_1 \mathcal{I} + \beta_2 \nabla_x^T \nabla_x - \beta_3 \nabla_y^T \nabla_y - \beta_4 \nabla_t^T \nabla_t) \mathcal{R} = \\ \beta_1 \mathcal{Y}^{(t+1)} + \boldsymbol{\Lambda}_1^{(t)} + \nabla_x^T (\beta_2 \mathcal{S}^{(t+1)} + \boldsymbol{\Lambda}_2^{(t)}) \\ + \nabla_y^T (\beta_3 \mathcal{X}^{(t+1)} - \beta_3 \nabla_x \mathcal{O} + \boldsymbol{\Lambda}_3^{(t)}) \\ + \nabla_t^T (\beta_4 \mathcal{T}^{(t+1)} - \beta_4 \nabla_t \mathcal{O}^{(t+1)} + \boldsymbol{\Lambda}_4^{(t)}). \end{aligned}$$

The solution has the following closed-form solution:

$$\mathcal{R}^{(t+1)} = \mathcal{F}^{-1} \left(\frac{\mathcal{F}(\mathcal{K}_1)}{\mathcal{F}(\mathcal{K}_2)} \right), \quad (16)$$

where \mathcal{F} and \mathcal{F}^{-1} denote the fast Fourier transform (FFT) and its inverse transform, respectively. Here

$$\begin{aligned} \mathcal{K}_1 = & \beta_1 \mathcal{Y}^{(t+1)} + \boldsymbol{\Lambda}_1^{(t)} + \nabla_x^T (\beta_2 \mathcal{S}^{(t+1)} + \boldsymbol{\Lambda}_2^{(t)}) + \nabla_y^T (\beta_3 \mathcal{X}^{(t+1)} \\ & - \beta_3 \nabla_x \mathcal{O} + \boldsymbol{\Lambda}_3^{(t)}) + \nabla_t^T (\beta_4 \mathcal{T}^{(t+1)} - \beta_4 \nabla_t \mathcal{O}^{(t+1)} + \boldsymbol{\Lambda}_4^{(t)}) \end{aligned}$$

and

$$\mathcal{K}_2 = \beta_1 \mathcal{I} + \beta_2 \nabla_x^T \nabla_x - \beta_3 \nabla_y^T \nabla_y - \beta_4 \nabla_t^T \nabla_t.$$

Multipliers updating: Finally, following the framework of the ADMM, the Lagrange multipliers $\boldsymbol{\Lambda} = [\boldsymbol{\Lambda}_1, \boldsymbol{\Lambda}_2, \boldsymbol{\Lambda}_3, \boldsymbol{\Lambda}_4]$ are updated as:

$$\begin{cases} \boldsymbol{\Lambda}_1^{(t+1)} = \boldsymbol{\Lambda}_1^{(t)} + \beta_1 (\mathcal{Y}^{(t+1)} - \mathcal{R}^{(t+1)}), \\ \boldsymbol{\Lambda}_2^{(t+1)} = \boldsymbol{\Lambda}_2^{(t)} + \beta_2 (\mathcal{S}^{(t+1)} - \nabla_x \mathcal{R}^{(t+1)}), \\ \boldsymbol{\Lambda}_3^{(t+1)} = \boldsymbol{\Lambda}_3^{(t)} + \beta_3 (\mathcal{X}^{(t+1)} - \nabla_y (\mathcal{O} - \mathcal{R}^{(t+1)})), \\ \boldsymbol{\Lambda}_4^{(t+1)} = \boldsymbol{\Lambda}_4^{(t)} + \beta_4 (\mathcal{T}^{(t+1)} - \nabla_t (\mathcal{O} - \mathcal{R}^{(t+1)})). \end{cases} \quad (17)$$

The proposed algorithm is summarized in Algorithm 1. Since the proposed model is convex, the convergence of the proposed algorithm is theoretically guaranteed under the ADMM framework [32].

Algorithm 1 Algorithm for video rain streak removal

Input: The rainy video \mathcal{O} ;

- 1: Initialization: $\mathcal{B}^{(0)} = \mathcal{O}$, $\mathcal{R}^{(0)} = \text{zeros}(m \times n \times t)$;
- 2: **while** not converged **do**
- 3: Update \mathcal{Y} via (11);
- 4: Update \mathcal{S} via (13), \mathcal{X} via (14), and \mathcal{T} via (15);
- 5: Update \mathcal{R} via (16);
- 6: Update the multipliers via (17);
- 7: **end while**

Output: The estimation of rain streaks \mathcal{R} and the clean video $\mathcal{B} = \mathcal{O} - \mathcal{R}$.

4. EXPERIMENTAL RESULTS

Preprocessing: The color video is a four-mode tensor of size $m \times n \times 3 \times t$. We convert videos from the RGB color space to YUV color space and only conduct the method on the Y channel. Thus the videos that we process become a three-mode tensor of size $m \times n \times t$. To reduce the boundary effect, we pad the input tensors $\mathcal{O} \in \mathbb{R}^{m \times n \times t}$ by 5-pixel-width under reflective boundary condition. Thus the size of the input tensors becomes $(m+10) \times (n+10) \times (t+10)$. To validate the effectiveness of the proposed method, we compare the proposed method with two state-of-the-art methods: rain streak removal using temporal correlation and low-rank matrix completion (LRMC) [6] and rain streak removal using discriminatively intrinsic priors (DIP) [23]. Readers can find the Matlab code (p-code) to test the performance of our method [there](#).

4.1. Synthetic data

For synthetic data, since the clean videos are available, the *peak signal to noise ratio* (PSNR) and *structure similarity* (SSIM) [33] are selected to measure the performance of methods. Six videos named as “carphone”, “container”, “coast-guard”, “bridgefar”, “highway” and “foreman”¹ are selected as our test datasets. These videos can be viewed as four-mode tensors of size $144 \times 176 \times 3 \times 150$.

Rainy videos generation: The rainy videos are generated by the following steps. (1) The salt and pepper noise is added to a zero tensor with the same size as the clean video tensor. (2) The noise tensor is blurred by Gaussian blur. (3) The blurred and noisy tensor is further blurred by motion blur. There exists 5-15

¹<http://trace.eas.asu.edu/yuv/>.

degrees between motion direction and vertical direction. (4) Finally, the blurred and noisy tensor is directly added to the clean videos, and the intensity values greater than 1 are set as 1.

Parameters setting: The parameters $\{\beta_1, \beta_2, \beta_3, \beta_4\}$ are set as 50, and other parameters $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ are selected from $\{0.1, 0.3, 1, 3, 10, 30, 100, 300, 1000\}$. The stopping criterion is that the relative error of rain streaks is less than 5×10^{-3} or the iteration number is larger than 250.

Performance comparisons: We can observe from Table 1, that the proposed method significantly outperforms the companying methods in terms of PSNR val-
ues and SSIM values. For light and heavy rain, the proposed method achieves the highest PSNR and SSIM values except the last video for light rain streaks. In average, the PSNR values of the proposed method are 8.016 dB and 2.966 dB higher than those of LRMC and DIP for heavy rain streaks. In average, the PSNR values of the proposed method are 7.292 dB and 0.330 dB higher than those of LRMC and DIP for light rain streaks.

Moreover, the frames of estimated videos are displayed in Figures 6 and 7 for visual inspection. As observed, the proposed method achieves significantly better visual quality than the compared methods in rain streak removal, visibility enhancement, and detail preservation. There are two main reasons. The first reason is that LRMC and DIP both assume the clean video is low-rank, which leads to that some obvious details are lost. However, we disuse the low-rankness of the clean video, which preserves the details in dynamic clean video. For example, DIP and LRMC remove the street lights in “highway” for both heavy rain streaks and light rain streaks . In “bridgefar”, although the clean video is almost static, some small objects such as water pattern destroy the low-rankness. Thus, the details of water pattern are lost in the results of DIP and LRMC. Another reason is that we use the group sparsity to characterize rain streaks, which helps to preserve the line pattern and keep the continuity of the rain streaks, leading to more accurate rain streak removal results than other methods. In comparison, DIP does not extract sufficient rain streaks and does not preserve the continuity of rain streaks, e.g., “coastguard” and “foreman” for heavy rain streaks and “carphone” for light rain streaks. Since the continuity is more significant for heavy rain streaks, the proposed method equipped with group sparsity term outperforms the companying methods for heavy rain streaks.

Discussion of each term: We investigate the role of each term in our model (7) by changing one parameter while fixing the others. Figure 8 shows the PSNR curves of the proposed method using different parameter settings, where the testing parameter is chosen from the geometric series $\{0.1, 0.121, \dots, 0.1 \times$

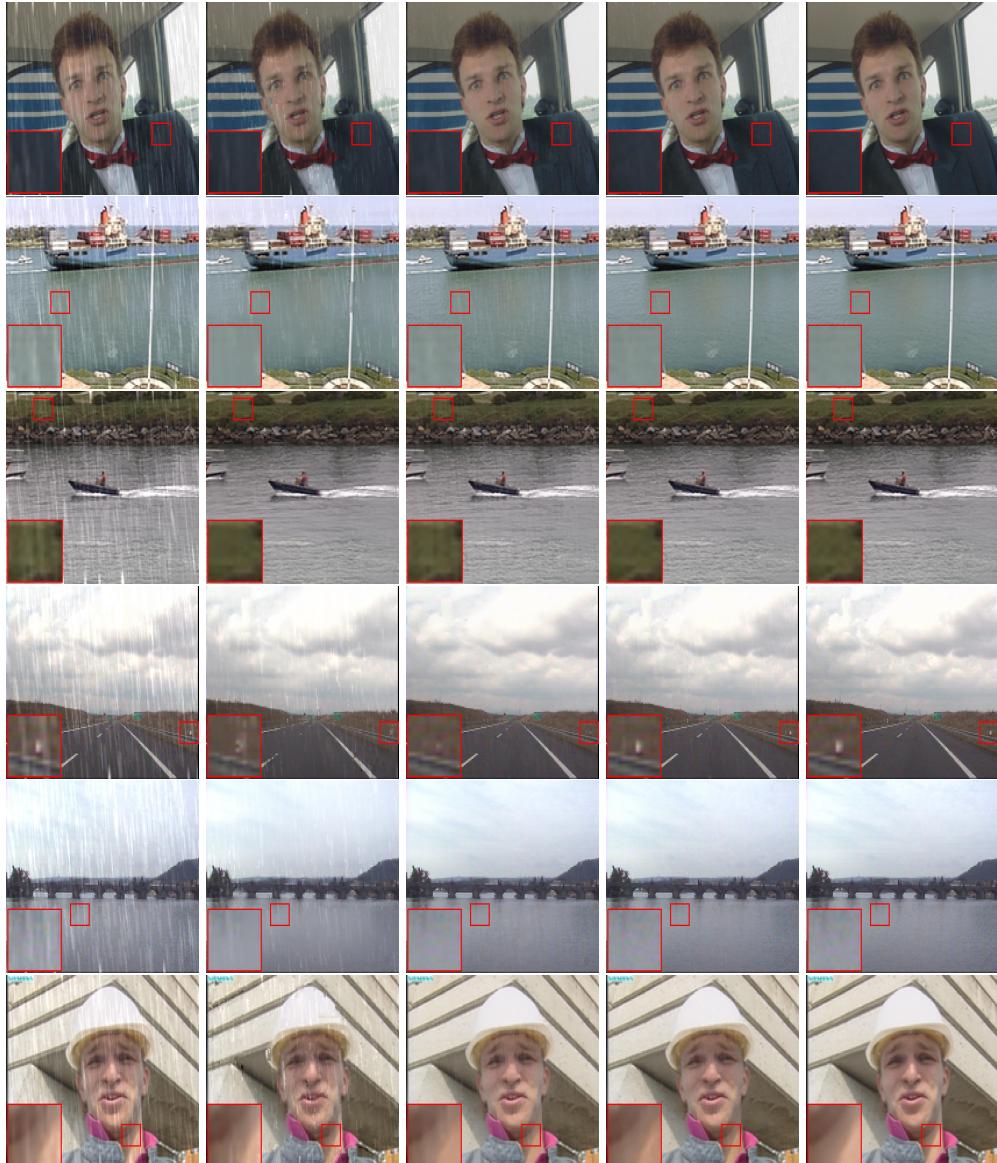


Figure 6: Rain streak removal results by different methods. From left to right: the rainy frames, the results by LRMC [6], DIP [23], the proposed method, and the ground truth. From top to bottom: the “carphone”, “container”, “coastguard”, “highway” “bridgefar” and “foreman” videos with the heavy synthetic rain streaks, respectively.

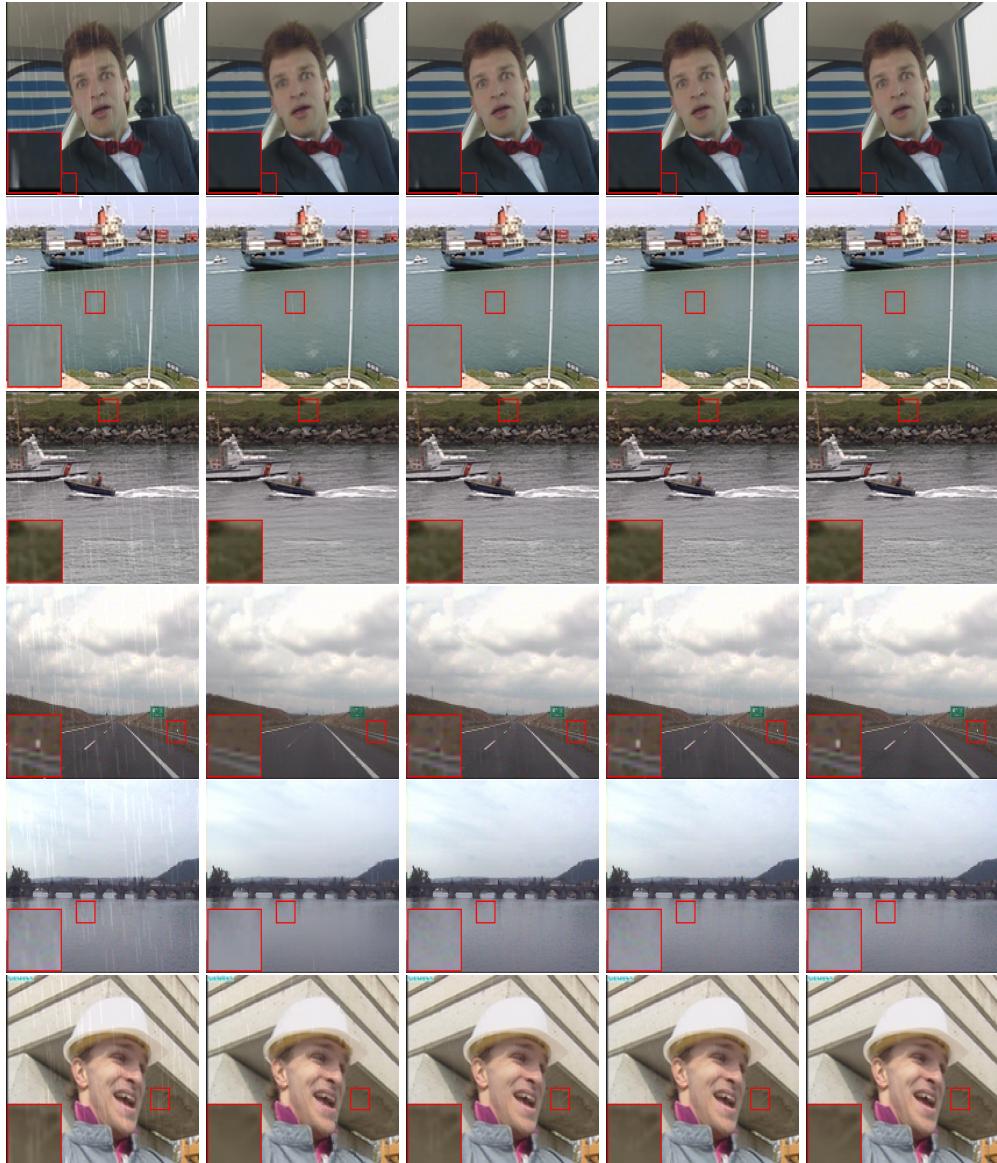


Figure 7: Rain streak removal results by different methods. From left to right: the rainy frames, the results by LRMC [6], DIP [23], the proposed method, and the ground truth. From top to bottom: the “carphone”, “container”, “coastguard”, “highway” “bridgefar” and “foreman” videos with the light synthetic rain streaks, respectively.

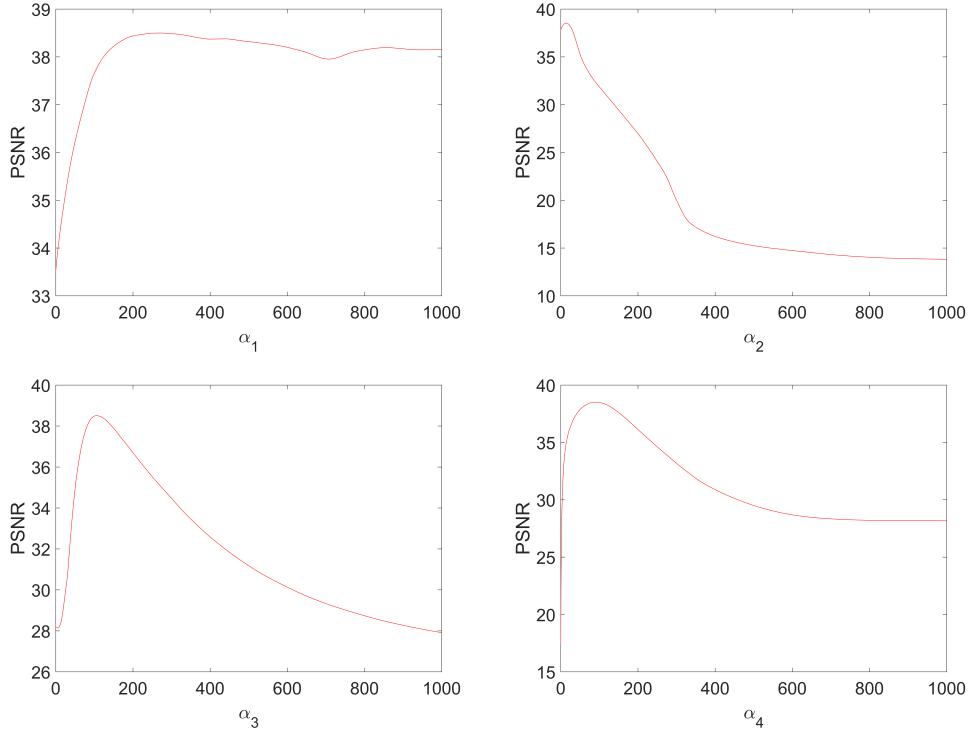


Figure 8: The PSNR values of the proposed method using different parameter settings.

$1.1^k, \dots, 1000\}$. It could be found that each parameter has an important contribution to the performance of the proposed method.

Discussion of groups: The group size is an vital important parameter which is set as one column in this paper unless otherwise specified. And it is very interesting to investigate the influence on the performance the proposed model with different group sizes. Table 2 shows the PSNR and SSIM values of the proposed model using different group sizes. From Table 2, we can observe that the group size has an impact on the performance of the proposed model. More specially, heavy videos favor large group sizes while light videos favor small group sizes. For simplicity, we choose one column as default in all experiments because there is no significant difference between different group sizes.

Discussions of the oblique rain streaks: Generally, the rain drops are falling from top to bottom and the rain streaks are close to being vertical. As we exhibited above, our method is robust to a small range of the angles since the rain streaks in

the synthetic data are not strictly vertical. However, the assumption is not always established (the angle between the direction of the rain streaks and the vertical direction would be very large). The proposed model consists of 4 regularization terms, which simultaneously contribute to the rain streak removal. When the rain streaks are oblique, the one regularizer corresponding to the directional property and the group sparsity of the rain streaks would not be helpful. Nonetheless, the temporal and the horizontal continuity of the background still exist. Thus, tuning the parameters to enlarge the effects of these two regularizers would help the proposed method to remove the rain streaks. Figures 9 and Table 3 show the results on two synthetic videos the “highway2”(35-55 degrees between the direction of the rain streaks and the vertical direction) and the “waterfall” (15-35 degrees between the direction of the rain streaks and the vertical direction) with oblique rain streaks. It can be found that when the rain streaks are not vertical, our method still works and achieves promising performances.



Figure 9: Rain streak removal results by different methods. From left to right: the rainy frames, the results by LRMC [6], DIP [23], the proposed method, and the ground truth. From top to bottom: the “highway2” and “waterfall” videos, respectively.

Discussions of the preprocessing: Before applying our algorithm, there are two preprocessing steps, i.e., (a) the conversion from RGB space to YUV space, and (b) adding reflective boundary condition. We would like to illustrate the influence of the two preprocessing steps using the video, “carphone”, with heavy rain streaks and light rain streaks. Table 4 shows the quantitative effects from these two preprocessing steps. It can be found from Table 4 that our algorithm generated comparative results with and without the conversion from RGB space to YUV space. This conversion would largely reduce the running time and hardly affect the performance. Meanwhile, as we expected, the reflective boundary condition slightly improved the performance. The method in [6] is designed for the RGB

videos so that we fed the RGB videos to it. The algorithm in [23] is also a tensor based method and involves the fast calculation using Fourier transform. For fair comparison, we did the same preprocessing steps when running the algorithm in [23]. It can be found that without the two preprocessing steps, the proposed method still work best.

4.2. Real data

We test two real rainy videos. One is a clipped part of size $260 \times 440 \times 3 \times 128$ from the movie “the Matrix”, and the other one is a backyard video of size $512 \times 256 \times 3 \times 128$ recorded in a rainy day. It is worth mentioning that the proposed method is not sensitive to parameters. The parameters for real data are the same as those in the first synthetic experiments.

Performance comparisons: For the first real video, we compare all the methods on one extreme cases. The first video is a very challenge video under lightning which enlarges the difference between adjacent frames and breaks the continuity along time direction. The rain streak removal results are displayed in Figure 10. And we can observe from Figure 10 that the rain streaks are more effectively removed by the proposed methods as compared with the other methods.

For the second real video, the rain streak removal results are displayed in Figure 11. We observe from Figure 11 that due to the clean video is static, which makes low-rankness a good video description, DIP performs well for this video. In spite of this, the rain streaks are more effectively removed by the proposed methods as compared with the other methods.

5. CONCLUSIONS

In this paper, we propose a tensor-based rain streak removal model. We use the group sparsity and the smoothness along the vertical direction to characterize rain streaks, and use the smoothness along the horizontal direction of rain streaks and the time direction to characterize the clean video. Meanwhile we discuss low-rankness. We develop an efficient ADMM algorithm to solve the proposed model. The experiments on synthetic and real data demonstrate the superiority of the proposed method over state-of-the-art method in terms of both quantitative and qualitative assessments. We will explore the group sparsity of the derivatives in the vertical direction of the rain streaks in our further work.

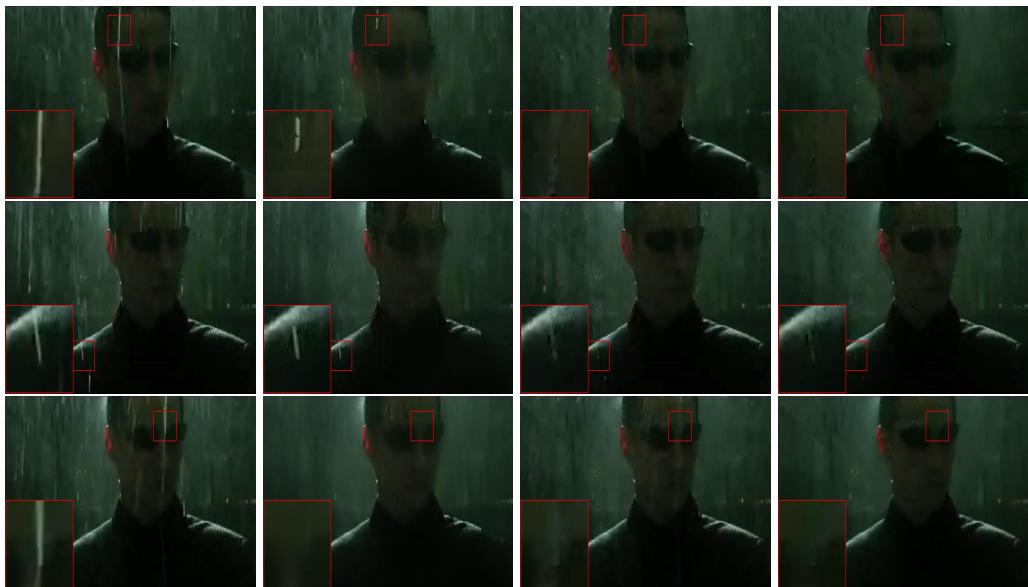


Figure 10: Rain streak removal results by different methods. From left to right: the rainy frames, the results by LRMC[6], DIP [23], and the proposed method. From top to bottom: three frames of the first real video.

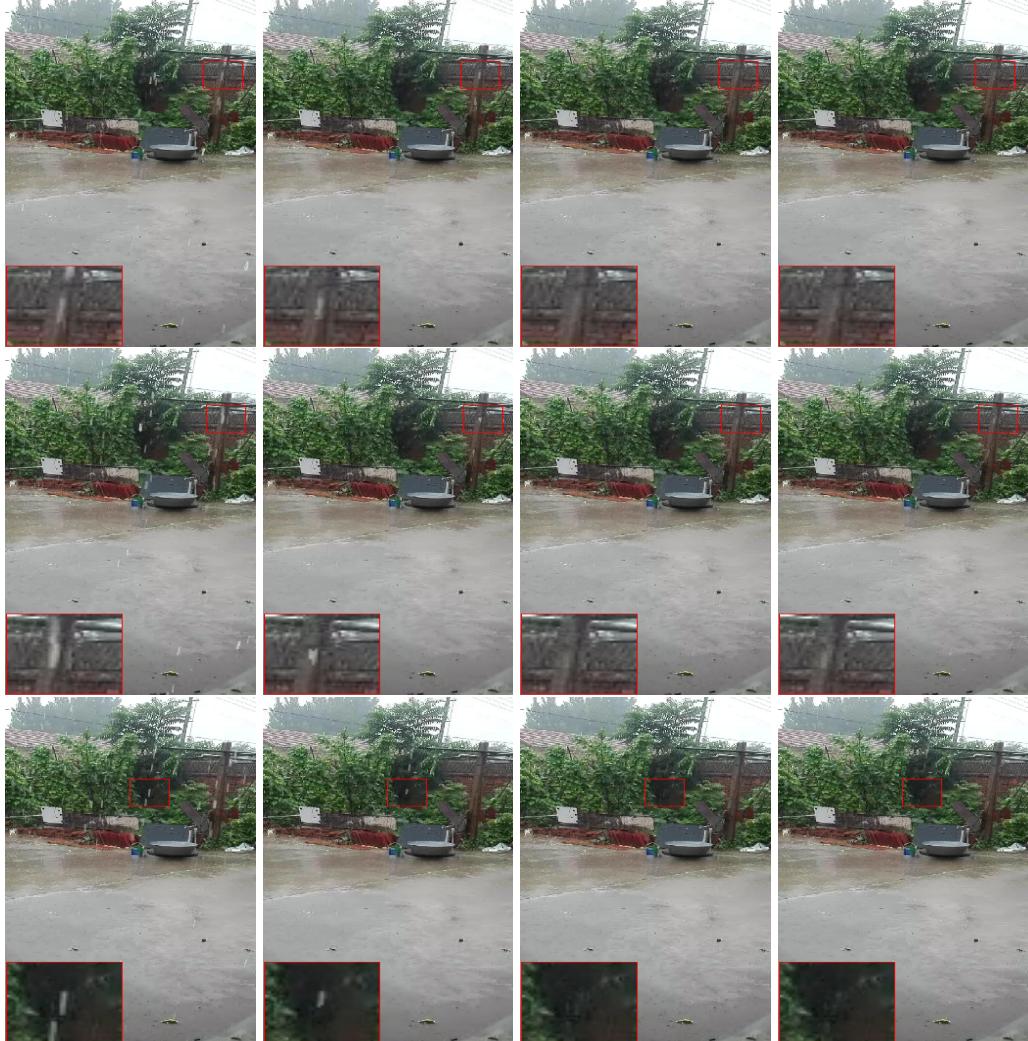


Figure 11: Rain streak removal results by different methods. From left to right: the rainy frames, the results by LRMC[6], DIP [23], and the proposed method.

ACKNOWLEDGEMENT

The research is supported by NSFC (61876203 61772003, 61702083) and the Fundamental Research Funds for the Central Universities (ZYGX2016J132, ZYGX2016KYQD142, ZYGX2016J129).

- [1] S. Maji, A.-C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. *In Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.
- [2] O.-L. Junior, D. Delgado, V. Goncalves, and U. Nunes. Trainable classifier-fusion schemes: An application to pedestrian detection. *In Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2009.
- [3] M.-S. Shehata, J. Cai, W.-M. Badawy, T.-W. Burr, M.-S. Pervez, R.-J. Johannesson, and A. Radmanesh. Video-based automatic incident detection for smart roads: the outdoor environmental challenges regarding false alarms. *IEEE Transactions on Intelligent Transportation Systems*, 9:349–360, 2008.
- [4] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:564–577, 2003.
- [5] K. Garg and S.-K. Nayar. Vision and rain. *International Journal of Computer Vision*, 75:3–27, 2007.
- [6] 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*, 24:2658–2670, 2015.
- [7] 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*, 21:1742–1755, 2012.
- [8] Y. Luo, Y. Xu, and H. Ji. Removing rain from a single image via discriminative sparse coding. *In Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3397–3405, 2015.

- [9] M. Roser, J. Kurz, and A. Geiger. Realistic modeling of water droplets for monocular adherent raindrop recognition using bezier curves. *In Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 235–244, 2011.
- [10] S.-D. You, R.-T. Tan, R. Kawakami, Y. Mukaigawa, and K. Ikeuchi. Adherent raindrop modeling, detectionand removal in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:1721–1733, 2016.
- [11] L.-J. Deng, T.-Z. Huang, X.-L. Zhao, and T.-X. Jiang. Rain removal from a single image via a unidirectionally global sparse model. *Applied Mathematical Modeling*, 2017.
- [12] 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 (ICCV)*, pages 1968–1975, 2013.
- [13] Y. Li, R.-T. Tan, X.-J. Guo, J.-B. Lu, and M.-S. Brown. Rain streak removal using layer priors. *In Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2736–2744, 2016.
- [14] S.-H. Sun, S.-P. Fan, and Y.-C.-F. Wang. Exploiting image structural similarity for single image rain removal. *In Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 4482–4486, 2014.
- [15] L. Zhu, C.-W. Fu, D. Lischinski, and P.-A. Heng. Joint bi-layer optimization for single-image rain streak removal. *In Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2545–2553, 2017.
- [16] Y. Chang, L.-X. Yan, and S. Zhong. Transformed low-rank model for line pattern noise removal. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1726–1734, 2017.
- [17] W.-H. Yang, R.-T. Tan, J.-S. Feng, J.-Y. Liu, Z.-M. Guo, and S.-C. Yan. Deep joint rain detection and removal from a single image. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1685–1694, 2017.
- [18] X.-Y. Fu, J.-B. Huang, X.-H. Ding, Y.-H. Liao, and J. Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, 26:2944–2956, 2017.

- [19] K. Garg and S.-K. Nayar. Detection and removal of rain from videos. *In Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 528–535, 2004.
- [20] A.-K. Tripathi and S Mukhopadhyay. Video post processing: low-latency spatiotemporal approach for detection and removal of rain. *IET image processing*, 6:181–196, 2012.
- [21] A.-K. Tripathi and S. Mukhopadhyay. Removal of rain from videos: a review. *Signal, Image and Video Processing*, 8:1421–1430, 2014.
- [22] W. Wei, L.-X. Yi, Q. Xie, Q. Zhao, D.-Y. Meng, and Z.-B. Xu. Should we encode rain streaks in video as deterministic or stochastic? *In Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2535–2544, 2017.
- [23] 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 Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 4057–4066, 2017.
- [24] X.-L. Zhao, W. Wang, T.-Y. Zeng, T.-Z. Huang, and M.-K. Ng. Total variation structured total least squares method for image restoration. *SIAM Journal on Scientific Computing*, 35:1304–1320, 2013.
- [25] X.-L. Zhao, F. Wang, and M.-K. Ng. A new convex optimization model for multiplicative noise and blur removal. *SIAM Journal on Imaging Sciences*, 7:456–475, 2014.
- [26] L.-J. Deng, W.-H. Guo, and T.-Z. Huang. Single image super-resolution by approximated heaviside functions. *Information Sciences*, 348:107–123, 2016.
- [27] L.-J. Deng, W.-H. 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*, 26:2001–2014, 2016.
- [28] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, T.-Y. Ji, and L.-J. Deng. Matrix factorization for low-rank tensor completion using framelet prior. *Information Sciences*, 2018. Accepted.

- [29] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang. Fastderain: A novel video rain streak removal method using directional gradient priors. *arXiv preprint arXiv:1803.07487*, 2018.
- [30] W. Deng, W.-T. Yin, and Y. Zhang. Group sparse optimization by alternating direction method. In *Proceedings of the The International Society of Optics and Photonics (SPIE)*, 8858, 2013.
- [31] K. Dan. A singularly valuable decomposition: The svd of a matrix. *College Mathematics Journal*, 27:2–23, 1996.
- [32] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations & Trends in Machine Learning*, 3:1–122, 2011.
- [33] 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*, 13:600–612, 2004.

Table 1: Quantitative comparisons of rain streak removal results by LRMC [6], DIP [23], and the proposed method, on the selected 6 synthetic videos, respectively.

Video	Method	Rain type			Heavy			Light		
		PSNR	SSIM	TIME(S)	PSNR	SSIM	TIME(S)	PSNR	SSIM	TIME(S)
carphone	Rainy	28.151	0.751	-	36.641	0.926	-	36.641	0.926	-
	LRMC	30.496	0.848	2230.193	36.490	0.978	1381.876	36.490	0.978	1381.876
	DIP	35.196	0.955	190.997	42.742	0.987	280.895	42.742	0.987	280.895
	Proposed	38.486	0.971	230.311	43.021	0.991	343.444	43.021	0.991	343.444
container	Rainy	28.551	0.758	-	37.162	0.929	-	37.162	0.929	-
	LRMC	31.338	0.877	1850.684	37.426	0.982	1240.786	37.426	0.982	1240.786
	DIP	39.093	0.970	184.324	51.061	0.998	259.875	51.061	0.998	259.875
	Proposed	45.252	0.993	293.509	51.363	0.998	317.864	51.363	0.998	317.864
coastguard	Rainy	28.128	0.833	-	36.579	0.956	-	36.579	0.956	-
	LRMC	34.955	0.960	2709.774	34.880	0.955	1980.656	34.880	0.955	1980.656
	DIP	34.338	0.963	203.535	40.070	0.985	285.622	40.070	0.985	285.622
	Proposed	35.951	0.971	344.890	40.222	0.986	423.444	40.222	0.986	423.444
highway	Rainy	29.056	0.744	-	37.524	0.925	-	37.524	0.925	-
	LRMC	33.388	0.890	1752.019	38.511	0.968	1308.776	38.511	0.968	1308.776
	DIP	39.469	0.968	238.900	43.564	0.985	297.554	43.564	0.985	297.554
	Proposed	41.281	0.974	367.434	43.629	0.982	444.590	43.629	0.982	444.590
bridgefar	Rainy	28.945	0.713	-	37.264	0.910	-	37.264	0.910	-
	LRMC	34.392	0.900	1678.564	41.852	0.974	1298.344	41.852	0.974	1298.344
	DIP	42.221	0.979	186.909	48.672	0.992	239.443	48.672	0.992	239.443
	Proposed	45.743	0.983	333.867	49.921	0.994	397.441	49.921	0.994	397.441
foreman	Rainy	28.341	0.808	-	36.954	0.947	-	36.954	0.947	-
	LRMC	30.101	0.855	2200.713	36.300	0.974	1460.754	36.300	0.974	1460.754
	DIP	34.650	0.965	190.546	41.122	0.988	254.388	41.122	0.988	254.388
	Proposed	36.050	0.967	289.332	41.055	0.987	338.564	41.055	0.987	338.564

Table 2: Quantitative comparisons of rain streak removal results by the proposed method with one column, half of one column, quarter of one column, eighth of one column.

Video	Rain type Method	Heavy			Light		
		PSNR	SSIM	TIME(S)	PSNR	SSIM	TIME(S)
carphone	Rainy	28.151	0.751	-	36.641	0.926	-
	one column	38.486	0.971	230.311	43.021	0.991	343.444
	half of one column	38.138	0.973	224.136	41.372	0.990	330.496
	quarter of one column	37.486	0.973	234.334	42.248	0.991	344.667
	eighth of one column	35.166	0.956	242.899	43.081	0.991	339.799
container	Rainy	28.551	0.758	-	37.162	0.929	-
	one column	45.252	0.993	293.509	51.363	0.998	317.864
	half of one column	45.146	0.992	289.778	51.900	0.998	328.565
	quarter of one column	44.677	0.991	288.526	52.347	0.998	331.965
	eighth of one column	44.837	0.993	299.657	52.430	0.998	329.999
coastguard	Rainy	28.128	0.833	-	36.579	0.956	-
	one column	35.951	0.971	344.890	40.222	0.986	423.444
	half of one column	35.982	0.970	339.756	40.538	0.986	434.899
	quarter of one column	35.934	0.965	346.813	40.665	0.987	423.131
	eighth of one column	35.754	0.970	334.287	40.497	0.986	435.998
highway	Rainy	29.056	0.744	-	37.524	0.925	-
	one column	41.281	0.974	367.434	43.629	0.982	444.590
	half of one column	39.899	0.970	359.142	43.326	0.986	435.827
	quarter of one column	41.799	0.976	339.982	43.413	0.985	437.896
	eighth of one column	41.842	0.977	378.869	43.223	0.983	447.867
bridgefar	Rainy	28.945	0.713	-	37.264	0.910	-
	one column	45.743	0.983	333.867	49.921	0.994	397.441
	half of one column	46.005	0.985	340.665	50.518	0.994	403.676
	quarter of one column	46.203	0.985	328.443	50.924	0.994	399.674
	eighth of one column	45.989	0.984	329.441	51.167	0.995	402.335
foreman	Rainy	28.341	0.808	-	36.954	0.947	-
	one column	36.050	0.967	289.332	41.055	0.987	338.564
	half of one column	36.090	0.966	296.996	40.693	0.986	365.447
	quarter of one column	35.781	0.966	302.154	39.327	0.986	336.732
	eighth of one column	36.009	0.966	288.838	40.317	0.988	332.655

Table 3: Quantitative comparisons of rain streak removal results by LRMC [6], DIP [23], and the proposed method, on the selected 2 synthetic videos, respectively.

Video	Method	Rain video			Quantitative comparisons		
		PSNR	SSIM	TIME(S)	PSNR	SSIM	TIME(S)
highway2	Rainy	27.170	0.803	-			
	LRMC	27.640	0.878	2530.393			
	DIP	33.406	0.929	258.067			
	Proposed	36.783	0.953	343.453			
waterfall	Rainy	28.551	0.758	-			
	LRMC	31.338	0.877	1850.684			
	DIP	35.593	0.939	184.324			
	Proposed	37.782	0.960	293.509			

Table 4: Quantitative comparisons of rain streak removal results by LRMC [6], DIP [23], and the proposed method on the “carphone”synthetic videos, respectively.

Rain type Method	Heavy			Light		
	PSNR	SSIM	TIME(s)	PSNR	SSIM	TIME(s)
Rainy	28.151	0.751	-	36.641	0.926	-
LRMC	30.496	0.848	2230.193	36.490	0.978	1381.876
DIP	35.196	0.955	190.997	42.742	0.987	280.895
Proposed	38.486	0.971	230.311	43.021	0.991	343.444
Proposed without (a)	38.406	0.969	763.256	43.005	0.990	1027.011
Proposed without (b)	37.856	0.962	221.054	42.958	0.989	310.520