Signal Processing: Image Communication 73 (2019) 96–108

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

A total variation and group sparsity based tensor optimization model for video rain streak removal

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

[a] School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, China
[b] School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, 710049, China

ARTICLE INFO

Keywords:
Video rain streak removal
Group sparsity
Unidirectional total variation (UTV)
Tensor optimization model
Alternating direction method of multipliers (ADMM)

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 the rain streaks and the clean video. In specific, the rain streaks are group sparse and smooth along the rain streaks direction; the clean video is smooth along the perpendicular direction of the rain streaks and the temporal direction. For the rain streaks, we use $\ell_{2,1}$ norm to characterize the group sparsity and the *unidirectional total variation* (UTV) to promote the smoothness along the rain streaks direction. For the clean video, we use two UTV regularizers to enhance the smoothness along the perpendicular direction of the rain streaks and the temporal direction. To solve the proposed model we develop an efficient *alternating direction method of multipliers* (ADMM) algorithm. Experiments on synthetic and real data demonstrate the superiority of the proposed method over state-of-the-art methods both quantitatively and qualitatively.

1. Introduction

Bad weather impairs the visibility of an image and introduces undesirable interference that can severely hinder the follow-up processing (e.g., object detection, recognition, and tracking [1–6]). This paper mainly focuses on the rain streak removal problem [7–12].

A single rainy image is generally modeled as $\mathbf{O} = \mathbf{B} + \mathbf{R}$ [8,13,14], 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 [15]: $\mathcal{O} = \mathcal{B} + \mathcal{R}$, where $\mathcal{O} \in \mathbb{R}^{m\times n\times t}$, $\mathcal{B} \in \mathbb{R}^{m\times n\times t}$, and $\mathcal{R} \in \mathbb{R}^{m\times n\times t}$ are the observed rainy video, the underlying clean video, and the rain streaks, respectively. The goal of the rain streak removal is to estimate the clean image/video from the rainy image/video. This typical inverse problem is often solved by regularization methods which incorporate additional prior knowledge.

In general, rain streak removal methods can be categorized into two classes: the single image rain streak removal methods and video rain streak removal methods. Traditional model driven methods for the single image rain streaks removal are usually designed by exploring prior knowledge to tackle the ill-posed inverse problem. Kang et al. [8] first decomposed an image into the low- and high-frequency parts and separates the rain streaks from the high-frequency part by performing dictionary learning and sparse coding. Sun et al. [16] proposed to explore the structural similarity of the image bases. Luo et al. [9] proposed a dictionary learning based method, which sparsely approximates the patches of two layers by very high discriminative codes over a learned dictionary. Based on Gaussian mixture models, Li et al. [14] used simple patch-based priors for both the clean and rain streaks. Zhu et al. [17] presented a joint bi-layer optimization model assisted by two clean image priors and one rain streaks prior. In [18], the directional property of rain streaks received attention. The recently emerged deep learning technique is also successfully applied to the single image rain streak removal task [19–27].

For the video rain streak removal, Garg et al. [28] firstly proposed a video rain streak removal method with a comprehensive analysis of the visual effects of rain streaks on an imaging system. Subsequently, a series of methods have been suggested for the video rain streak removal. Tripathi et al. [29] detected and removed rain streaks using spatio-temporal properties. Chen [13] proposed and generalized a low-rank model from matrix to tensor structure in order to capture the spatio-temporally correlated rain streaks. Kim et al. [7] obtained an initial rain map by exploring the temporal correlation and finally removed the detected rain streaks by using a low-rank matrix completion. Very recently, the rain streaks were modeled as a mixture of Gaussians distribution in [30]. In [15], a novel tensor-based video rain streak removal method was proposed by considering discriminative prior of the clean video and the rain streaks. For the video rain streak removal, the deep learning based methods also started to reveal their effectiveness [31,32]. The existing video-based methods are comprehensively reviewed in [33].

* Corresponding authors.
E-mail addresses: xlzhao122003@163.com (X.-L. Zhao), taixiangjiang@gmail.com (T.-X. Jiang).

https://doi.org/10.1016/j.image.2018.11.008
Received 2 March 2018; Received in revised form 25 November 2018; Accepted 26 November 2018
Available online 1 December 2018
0923-5965/

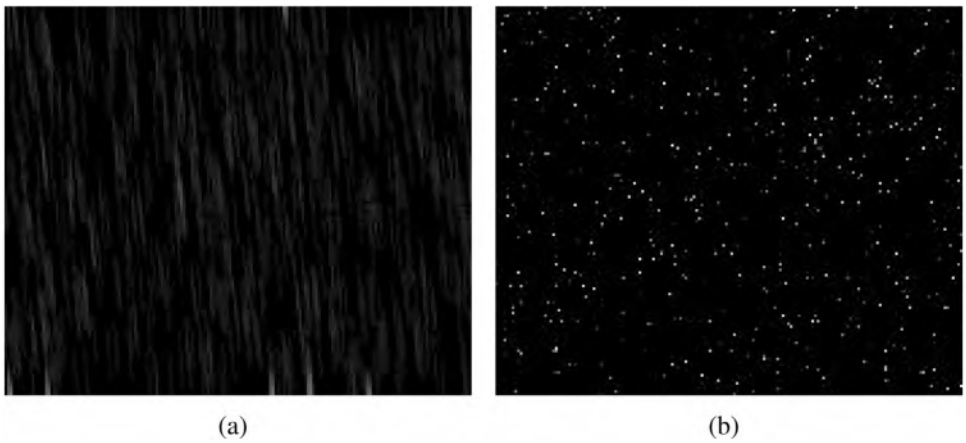

Fig. 1. (a) The rain image and (b) a random sparse image.

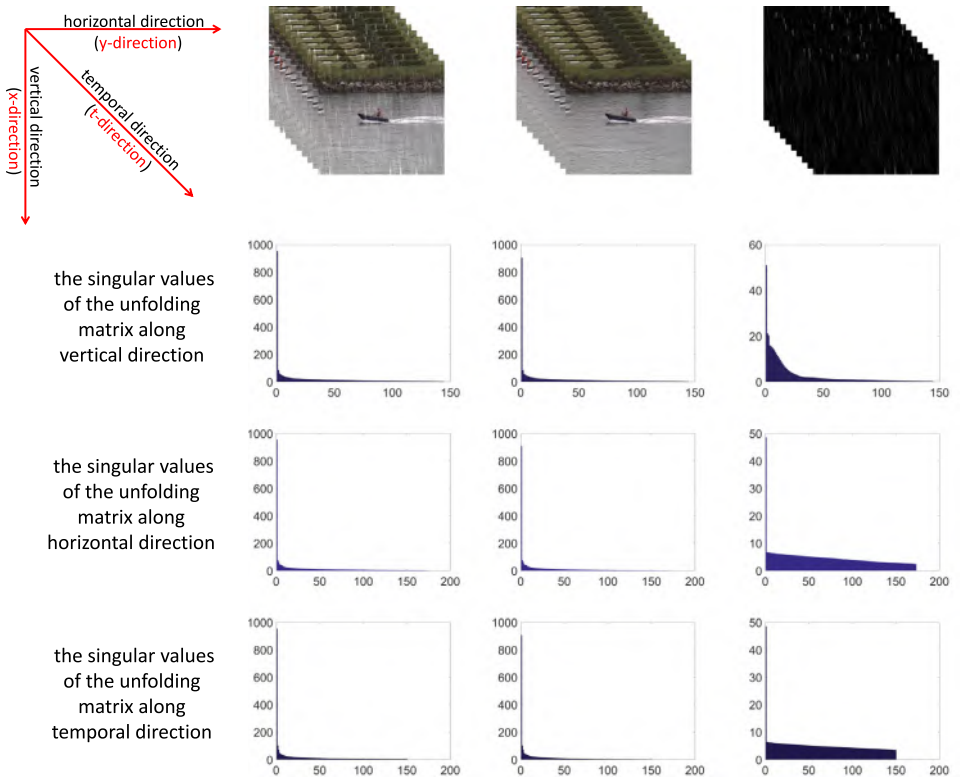

Fig. 2. From left to right: the singular values of unfolding matrices of the rainy video, the clean video, and the rain streaks.

In [15], Jiang et al. proposed the following model:

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

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

However, the model in (1) has two drawbacks. One is that the rain streaks are not only typically sparse but also exhibit structural line patterns; see Fig. 1. Thus, it is better to utilize the group sparsity regularizer to simultaneously enhance the sparsity and characterize the line pattern. The other one is that the clean video does not exhibit obvious low-rankness when the video has moving foreground or dynamic background; see Fig. 2. Boosting the low-rankness may be powerful [34–36], but it is not suitable for the rain streak removal in some cases. Hence, there is room for improvement. The above observations motivate us to introduce the group sparsity regularizer for the rain streaks while discarding the low-rankness regularizer for the clean video. The proposed tensor optimization model consists of the group sparsity regularizer and the *unidirectional total variation* (UTV) regularizer along the vertical direction for the rain streaks and the UTV regularizers along the horizontal direction and temporal direction for the clean video. In summary, the proposed model can be formulated as

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

To solve the proposed model, we develop an efficient *alternating direction method of multipliers* (ADMM) [37–40] algorithm. Experimental results demonstrate the superiority of the proposed method qualitatively and visually.

The rest of this paper is organized as follows. In Section 2, the notations and basic knowledge are introduced. In Section 3, the proposed model and the proposed algorithm are presented. Experimental results are reported in Section 4. Finally, Section 5 concludes this paper.

2. Tensor basics

Following [15,41–43], 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 matrices, and bold upper calligraphic letters (e.g., $\mathcal{X}$) for tensors. An n-mode tensor is denoted as $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$. Its $(i_1, i_2, \ldots, i_n)$th element is denoted as $x_{i_1,i_2,\ldots,i_n}$, where $1 \le i_k \le I_k$ and $1 \le k \le n$. The inner product of two same-sized tensors is defined as

$$\langle \mathcal{X}, \mathcal{Y} \rangle := \sum_{i_1,i_2,\ldots,i_n} x_{i_1,i_2,\ldots,i_n} \times y_{i_1,i_2,\ldots,i_n}. \tag{3}$$

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

$$\|\mathcal{X}\|_F := \langle \mathcal{X}, \mathcal{X} \rangle^{\frac{1}{2}} = \Big(\sum_{i_1,i_2,\ldots,i_n} |x_{i_1,i_2,\ldots,i_n}|^2 \Big)^{\frac{1}{2}}. \tag{4}$$

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

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

When $i_k = 1$, $i_k - 1$ is set to be I_k. The "unfold" operation along the kth mode on a tensor $\mathcal{X}$ is defined as

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

Its inverse operation "fold" is defined as

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

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 $1 \le k \le n$,

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

and

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

Given the index set g_i, $\mathbf{x}_{g_i}$ is the subvector of $\mathbf{x}$ indexed by g_i. For the vector $\mathbf{x}$, $\ell_{2,1}$ norm is defined as:

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

which is always used to promote group sparsity [44]. For the matrix $\mathbf{X}$, $\ell_{2,1}$ norm is defined as

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

where g_i is the ith column. We can extend $\ell_{2,1}$ norm to the tensor case

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

A more extensive overview of group sparsity can be found in [44].

3. The proposed method and algorithm

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

3.1. The 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:

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

where ∇_x, ∇_y, and ∇_t are the derivative operators along the vertical direction, the horizontal direction, and the temporal direction, respectively. In the following, we will explain the terms in our model in details.

Group sparsity of the rain streaks: When the rain is not extremely heavy, the rain component is sparser than the clean video, and the rain component exhibits structural line patterns rather than distributes randomly, as shown in Fig. 1. Therefore, we use the term $\|\mathcal{R}\|_{2,1}$ to characterize the group sparsity which enhances the sparsity and preserves the line pattern, simultaneously. It is superior over the plain the ℓ_1 regularizer that has been validated in [15].

The smoothness of the rain streaks along the vertical direction: Generally, the rain streaks share similar directions. We assume that the direction of the rain streaks is the vertical direction. The derivatives of rain streaks and the clean video along the vertical direction are different, i.e., the derivatives along the vertical direction of the rain streaks are more sparse compared with those of the clean video; see Fig. 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 of the clean video along the horizontal direction: Natural images are piecewise smooth [45,46], which indicates that the derivatives of frames in the video are not dense along vertical and horizontal directions. The vertical rain streaks destroy the smoothness along the horizontal direction. Compared with the derivatives of the rain video, the derivatives of the clean video are sparse along the horizontal direction. As a result, the derivatives along the horizontal direction of rain video are dense, which is shown in Fig. 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 of the clean video along the temporal direction: Since a video generally maintains at least 25 frames per second, there would be the strong smoothness along temporal direction. The derivatives of the clean video are sparse along the temporal direction. However, because of the high velocity, the rain streaks do not share this temporal continuity. As displayed in Fig. 5, the derivatives along the temporal direction of the clean video are sparse while those of the rain streaks are not sparse. Therefore, we use the ℓ_1 norm of $\nabla_t \mathcal{B}$ to enhance the smoothness along the temporal direction of the clean video.

Discussion of the low-rankness: In our model, we discard the low-rankness regularizer which is considered in [15]. From Fig. 2, we can observe the low-rankness of the clean video is not significant, because of the moving objects and dynamical backgrounds. Meanwhile, the rain streaks may be spatially low-rank due to their repeatability [18]. Therefore, we drop the low-rank regularization term in (1).

3.2. The proposed algorithm

The proposed model in (7) is a convex optimization problem which can be solved by various convex optimization algorithms. We develop an efficient ADMM algorithm to solve the proposed model. By introducing the auxiliary variables $\mathcal{Y}$, $\mathcal{S}$, $\mathcal{X}$, and $\mathcal{T} \in \mathbb{R}^{m \times n \times t}$, we rewrite the proposed model as the following equivalent constrained problem:

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

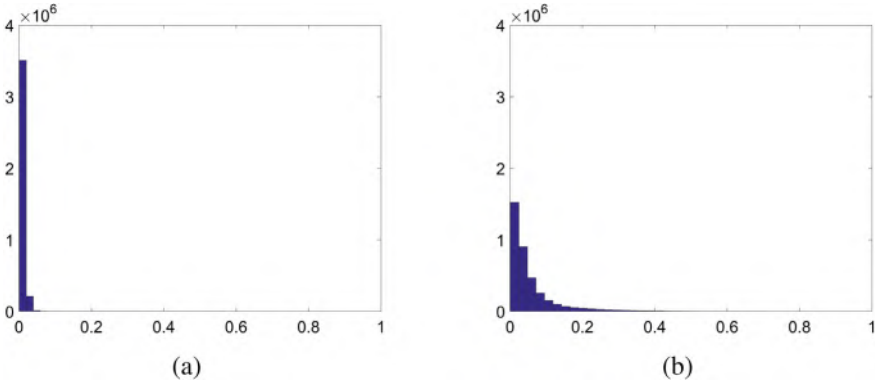

Fig. 3. (a) The histogram of the absolute values of the derivatives along the vertical direction of the rain streaks. (b) The histogram of the absolute values of the derivatives along the vertical direction of the clean video.

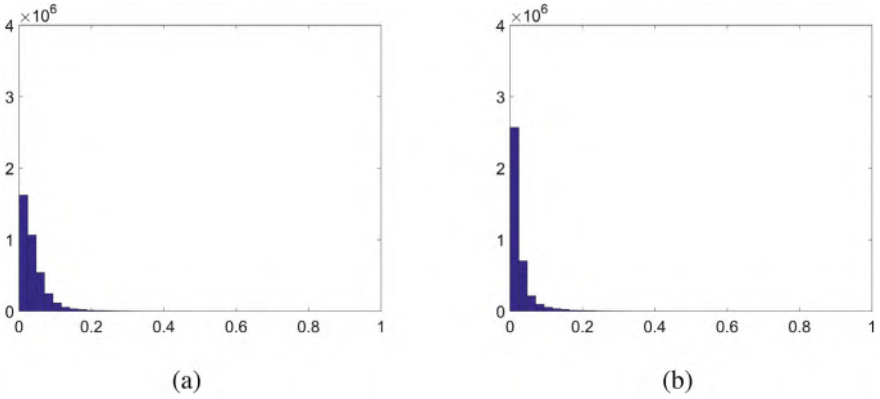

Fig. 4. (a) The histogram of the absolute values of the derivatives along the horizontal direction of the rain video. (b) The histogram of the absolute values of the derivatives along the horizontal direction of the clean video.

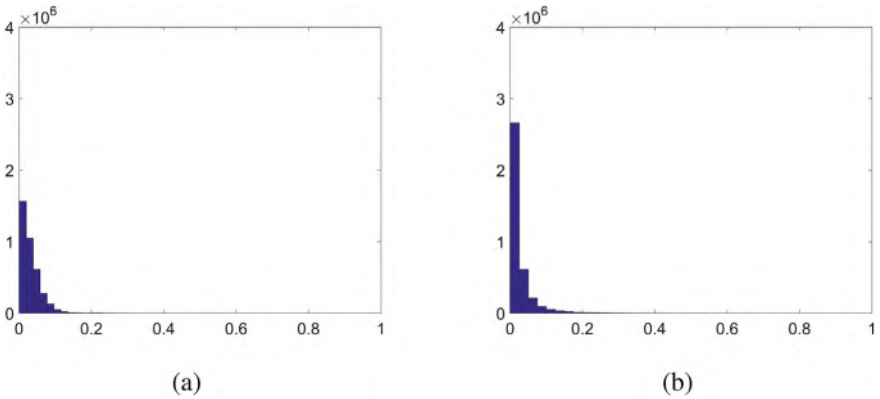

Fig. 5. (a) The histogram of the absolute values of the derivatives along the temporal direction of the rain streaks. (b) The histogram of the absolute values of the derivatives along the temporal direction of the clean video.

Then the augmented Lagrangian function of (8) is:

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

where $\mathbf{\Lambda}=[\mathbf{\Lambda}_1,\mathbf{\Lambda}_2,\mathbf{\Lambda}_3,\mathbf{\Lambda}_4]$ is Lagrange multiplier and β is a positive penalty parameter. This joint minimization problem can be decomposed into five subproblems which can be easily solved. By separating the variables in (9) into two groups $\mathcal{R}$ and $[\mathcal{Y},\mathcal{S},\mathcal{X},\mathcal{T}]$, the optimization problem in (9) fits the framework of ADMM. Then, the variables of two groups can be alternately updated. The solutions of the five subproblems are introduced in the following.

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

$$\arg\min_{\mathcal{Y}} \quad \alpha_1\|\mathcal{Y}\|_{2,1}+\frac{\beta}{2}\|\mathcal{Y}-\mathcal{R}+\frac{\mathbf{\Lambda_1}}{\beta}\|_F^2, \tag{10}$$

which is equivalent to

$$\arg\min_{\mathcal{Y}} \quad \sum_{g_i}(\alpha_1\|\mathcal{Y}_{g_i}\|_2+\frac{\beta}{2}\|\mathcal{Y}_{g_i}-\mathcal{Q}_{g_i}\|_F^2),\ \mathcal{Q}_{g_i}=\mathcal{R}_{g_i}-\frac{(\mathbf{\Lambda_1})_{g_i}}{\beta}.$$

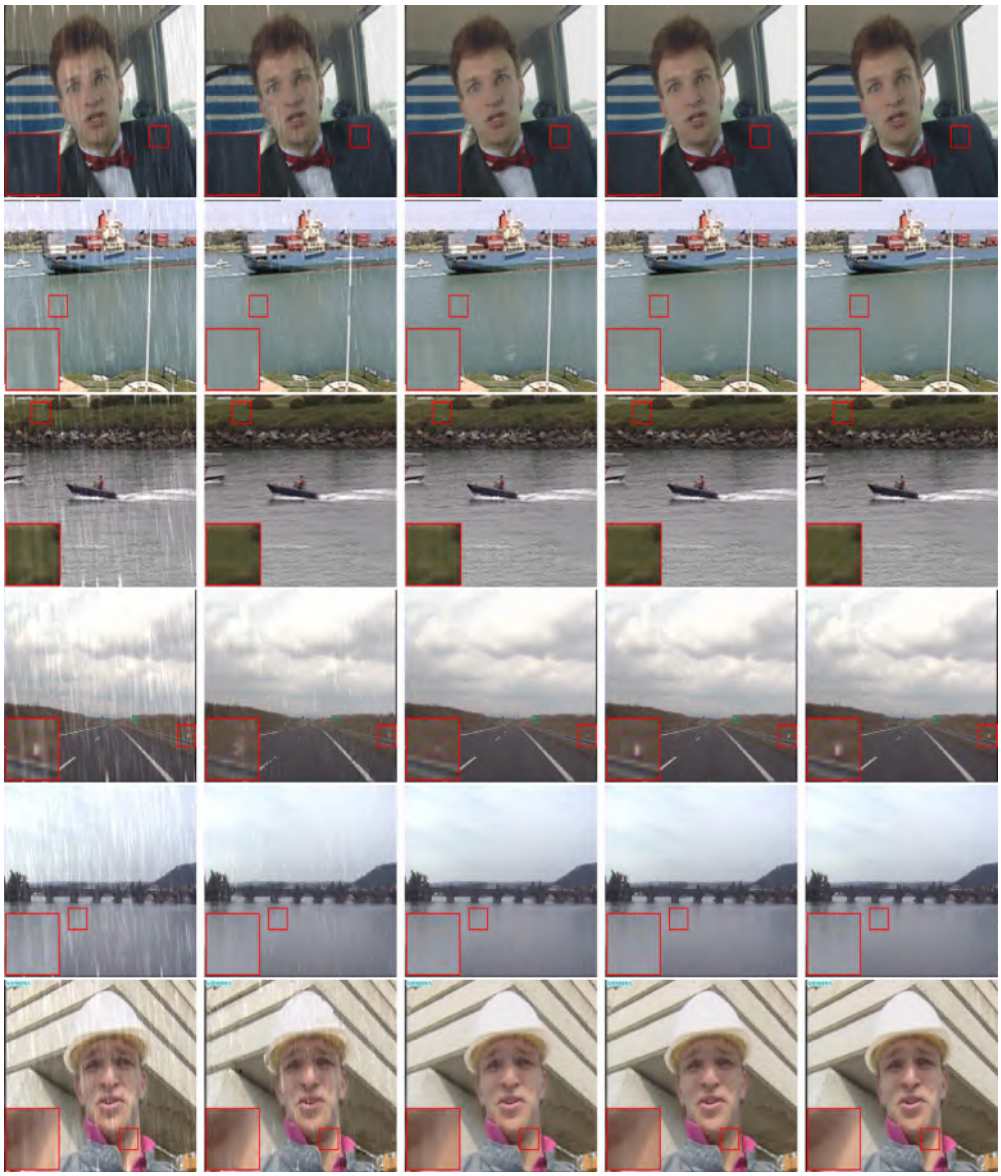

Fig. 6. Rain streak removal results by different methods. From left to right: the rainy frames, the results by LRMC [7], DIP [15], and the proposed method, and the ground truth. From top to bottom: the "carphone", "container", "coastguard", "highway" "bridgefar" and "foreman" videos with the heavy synthetic rain streaks.

Thus $\mathcal{Y}$ could be exactly updated via soft-shrinkage [44] as

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

where $\mathcal{Y}_{g_i}$ denotes the ith group of the variable $\mathcal{Y}$.

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

$$\begin{aligned}
&\arg\min_{\mathcal{S}} \quad \alpha_2\|\mathcal{S}\|_1 + \frac{\beta}{2}\|\mathcal{S} - \nabla_x\mathcal{R} + \frac{\mathbf{\Lambda_2}}{\beta}\|_F^2,\\
&\arg\min_{\mathcal{X}} \quad \alpha_3\|\mathcal{X}\|_1 + \frac{\beta}{2}\|\mathcal{X} - \nabla_y(\mathcal{O}-\mathcal{R}) + \frac{\mathbf{\Lambda_3}}{\beta}\|_F^2,\\
&\arg\min_{\mathcal{T}} \quad \alpha_4\|\mathcal{T}\|_1 + \frac{\beta}{2}\|\mathcal{T} - \nabla_t(\mathcal{O}-\mathcal{R}) + \frac{\mathbf{\Lambda_4}}{\beta}\|_F^2.
\end{aligned} \tag{12}$$

The elements of $\mathcal{S}$, $\mathcal{X}$, and $\mathcal{T}$ can be exactly updated by componentwise soft thresholding as

$$\mathcal{S}^{(t+1)} = \text{soft}_{\frac{\alpha_2}{\beta}}\left(\nabla_x\mathcal{R}^{(t)} - \frac{\mathbf{\Lambda}_2^{(t)}}{\beta}\right), \tag{13}$$

$$\mathcal{X}^{(t+1)} = \text{soft}_{\frac{\alpha_3}{\beta}}\left(\nabla_y(\mathcal{O}-\mathcal{R}^{(t)}) - \frac{\mathbf{\Lambda}_3^{(t)}}{\beta}\right), \tag{14}$$

$$\mathcal{T}^{(t+1)} = \text{soft}_{\frac{\alpha_4}{\beta}}\left(\nabla_t(\mathcal{O}-\mathcal{R}^{(t)}) - \frac{\mathbf{\Lambda}_4^{(t)}}{\beta}\right), \tag{15}$$

where

$$\text{soft}_{\frac{\alpha}{\beta}}(y) = \text{sign}(y)\max(|y| - \frac{\alpha}{\beta}, 0).$$

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

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

The normal equation of the least square problem is

$$\begin{aligned}
(\beta\mathcal{I} + \beta\nabla_x^{\mathrm{T}}\nabla_x + \beta\nabla_y^{\mathrm{T}}\nabla_y + \beta\nabla_t^{\mathrm{T}}\nabla_t)\mathcal{R} &= \beta\mathcal{Y} + \mathbf{\Lambda}_1 + \nabla_x^{\mathrm{T}}(\beta\mathcal{S} + \mathbf{\Lambda}_2)\\
&+ \nabla_y^{\mathrm{T}}(\beta\nabla_x\mathcal{O} - \beta\mathcal{X} - \mathbf{\Lambda}_3) + \nabla_t^{\mathrm{T}}(\beta\nabla_t\mathcal{O} - \beta\mathcal{T} - \mathbf{\Lambda}_4).
\end{aligned}$$

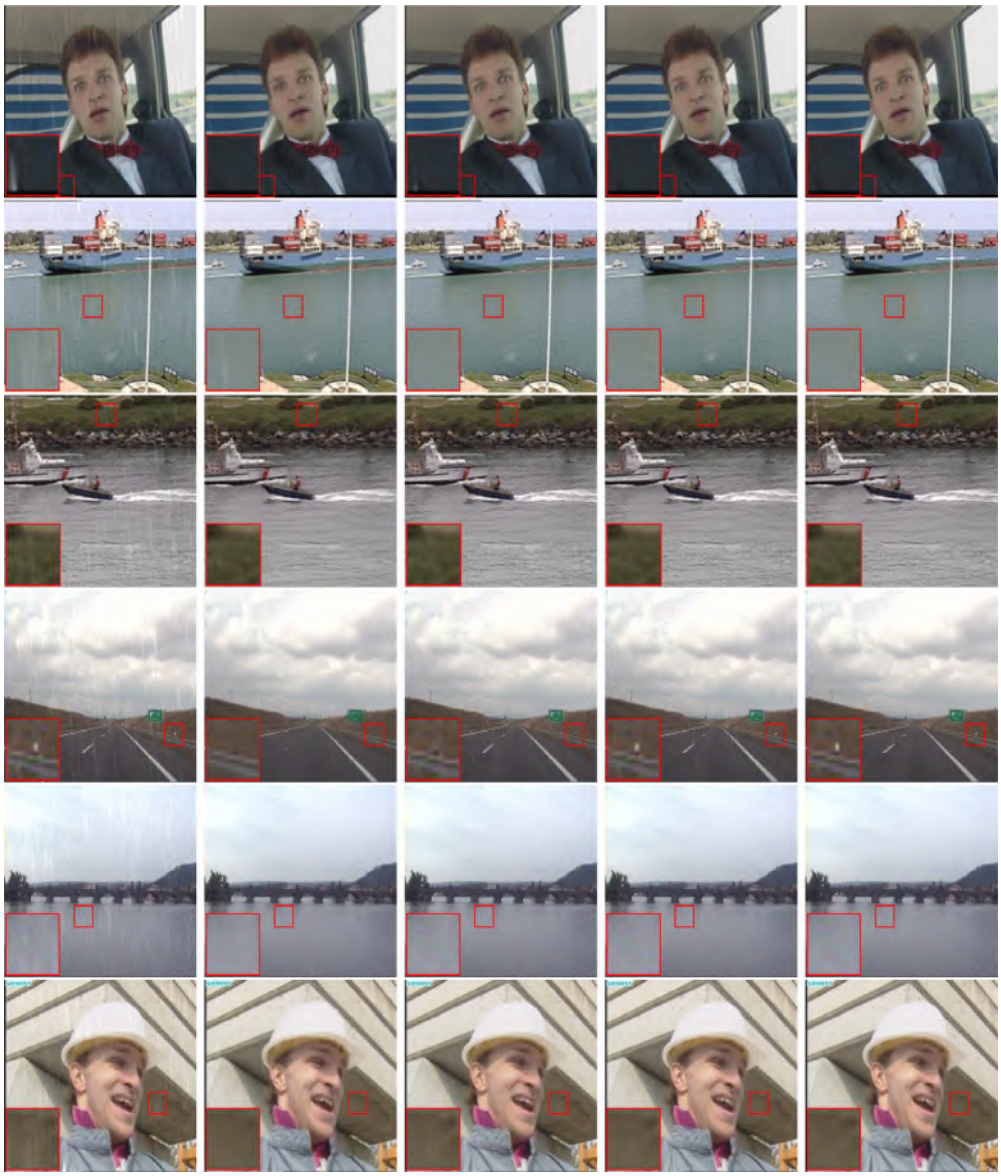

Fig. 7. Rain streak removal results by different methods. From left to right: the rainy frames, the results by LRMC [7], DIP [15], and the proposed method, and the ground truth. From top to bottom: the "carphone", "container", "coastguard", "highway" "bridgefar" and "foreman" videos with the light synthetic rain streaks.

Thus, the $\mathcal{R}$ subproblem has the following closed-form solution:

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

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

$$\begin{aligned}\mathcal{K}_1 =& \beta\mathcal{Y}^{(t+1)} + \mathbf{\Lambda}_1^{(t)} + \nabla_x^{\mathrm{T}}(\beta S^{(t+1)} + \mathbf{\Lambda}_2^{(t)}) + \nabla_y^{\mathrm{T}}(\beta\nabla_x\mathcal{O} \\ &- \beta\mathcal{X}^{(t+1)} - \mathbf{\Lambda}_3^{(t)}) + \nabla_t^{\mathrm{T}}(\beta\nabla_t\mathcal{O} - \beta\mathcal{T}^{(t+1)} - \mathbf{\Lambda}_4^{(t)})\end{aligned}$$

and

$$\mathcal{K}_2 = \beta\mathcal{I} + \beta\nabla_x^{\mathrm{T}}\nabla_x + \beta\nabla_y^{\mathrm{T}}\nabla_y + \beta\nabla_t^{\mathrm{T}}\nabla_t.$$

The element of $\mathcal{R}$ smaller than 0 is mapped to 0. And the element of $\mathcal{R}$ larger than the corresponding element of $\mathcal{O}$ is mapped to the corresponding element of $\mathcal{O}$.

Multipliers updating: Finally, the Lagrange multipliers $\mathbf{\Lambda} = [\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \mathbf{\Lambda}_3, \mathbf{\Lambda}_4]$ are updated as:

$$\begin{cases}\mathbf{\Lambda}_1^{(t+1)} = \mathbf{\Lambda}_1^{(t)} + \beta(\mathcal{Y}^{(t+1)} - \mathcal{R}^{(t+1)}), \\ \mathbf{\Lambda}_2^{(t+1)} = \mathbf{\Lambda}_2^{(t)} + \beta(S^{(t+1)} - \nabla_x\mathcal{R}^{(t+1)}), \\ \mathbf{\Lambda}_3^{(t+1)} = \mathbf{\Lambda}_3^{(t)} + \beta(\mathcal{X}^{(t+1)} - \nabla_y(\mathcal{O} - \mathcal{R}^{(t+1)})), \\ \mathbf{\Lambda}_4^{(t+1)} = \mathbf{\Lambda}_4^{(t)} + \beta(\mathcal{T}^{(t+1)} - \nabla_t(\mathcal{O} - \mathcal{R}^{(t+1)})).\end{cases} \tag{17}$$

The proposed algorithm is summarized in Algorithm 1 and the Matlab demo of the proposed method is available online.[1] Since the proposed model is convex and the variables can be separated into two groups, the convergence of the proposed algorithm is theoretically guaranteed under the ADMM framework [47].

[1] https://github.com/munaiyi719/A-Total-Variation-and-Group-Sparsity-Based-Tensor-Optimization-Model-for-Video-Rain-Streak-Removal.

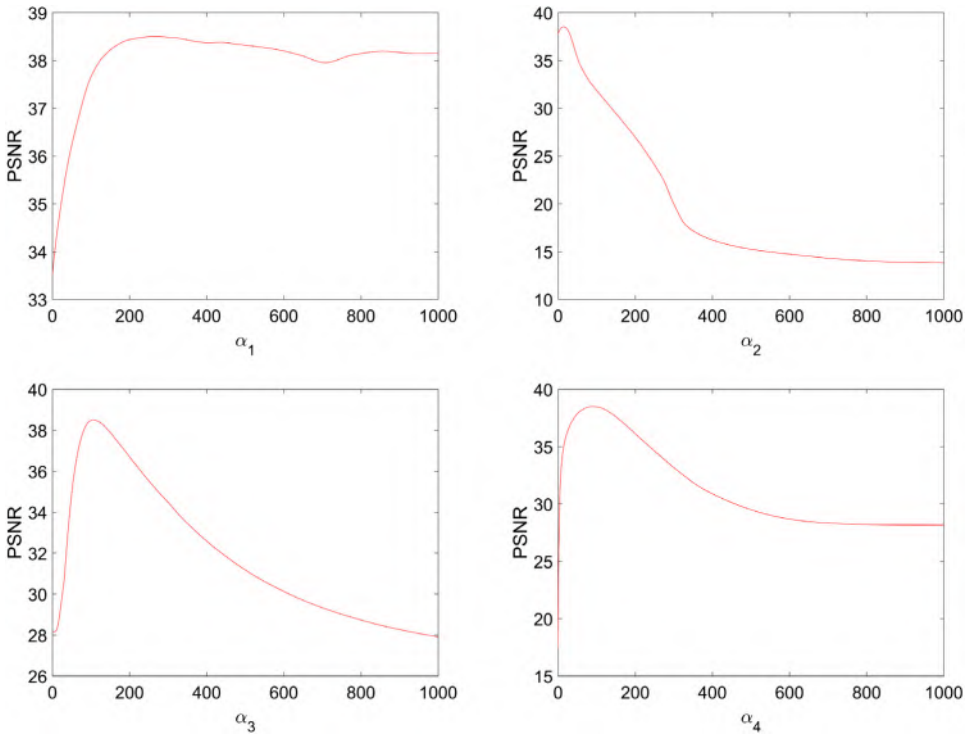

Fig. 8. The PSNR values of the proposed method using different parameter settings.

Table 1
Quantitative comparisons of rain streak removal results by LRMC [7], DIP [15], and the proposed method, on the selected 6 synthetic videos.

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

4. Experimental results

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) [7] and rain streak removal using discriminatively intrinsic priors (DIP) [15].

Algorithm 1 The proposed algorithm for video rain streak removal

Input: The rainy video $\mathcal{O}$;
1: Initialization: $\mathcal{B}^{(0)} = \mathcal{O}$, $\mathcal{R}^{(0)} = \mathcal{O} - \mathcal{B}^{(0)}$;
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 $\mathcal{B} = \mathcal{O} - \mathcal{R}$.

Preprocessing: The color video is a four-mode tensor of size $m \times n \times 3 \times t$. We convert the video from the RGB color space to YUV[2] color space and only conduct the proposed method on Y channel of YUV color space. Thus, the video is a three-mode tensor of size $m \times n \times t$. To reduce the boundary effect, we add 5 elements of padding to each dimension of a three-mode tensor under reflective boundary condition. Thus, the size of resulting input tensor is $(m + 10) \times (n + 10) \times (t + 10)$.

4.1. Synthetic data

For synthetic data, since the clean video is available, the *peak signal to noise ratio* (PSNR) and *structure similarity* (SSIM) [48] are selected to measure the performance of different methods. Six videos named as "carphone", "container", "coastguard", "bridgefar", "highway" and "foreman"[3] are selected as testing videos. These videos can be viewed as four-mode tensors of size $144 \times 176 \times 3 \times 150$.

Generation of rainy videos: 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] https://en.wikipedia.org/wiki/YUV.
[3] http://trace.eas.asu.edu/yuv/.

Fig. 9. Rain streak removal results by different methods. From left to right: the rainy frames, the results by LRMC [7], DIP [15], and the proposed method, and the ground truth. From top to bottom: the "highway2"and "waterfall" videos, respectively.

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

Rain type		Heavy			Light		
Video	Method	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 [7], DIP [15], and the proposed method, on the selected 2 synthetic videos, respectively.

Rain video		Quantitative comparisons		
Video	Method	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 [7], DIP [15], and the proposed method on the "carphone"synthetic videos, respectively.

Rain type	Heavy			Light		
Method	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

2. The noise tensor is blurred by a Gaussian blur.
3. The blurred and noisy tensor is further blurred by a motion blur. The angles between motion direction and vertical direction are randomly distributed in $[5°, 15°]$.
4. 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 parameter β is set as 50, and the parameters $\alpha_1, \alpha_2, \alpha_3$, and α_4 are selected from {0.1, 0.3, 1, 3, 10, 30, 100, 300, 1000}. The stopping criterion is that the relative error of the rain streaks

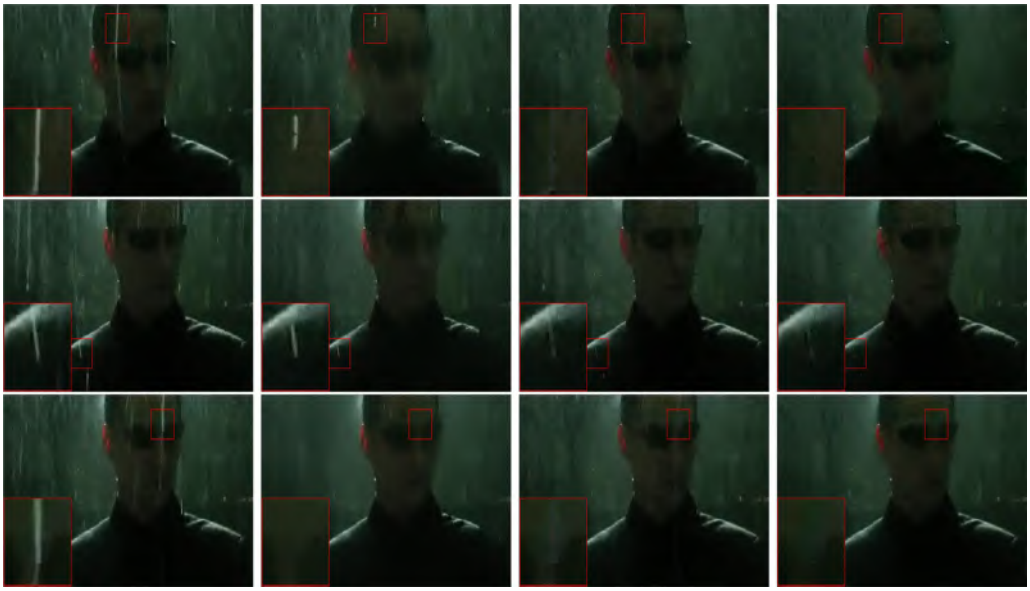

Fig. 10. Rain streak removal results of different methods on the video "the Matrix". From left to right: the rainy frames, the results by LRMC [7], DIP [15], and the proposed method. From top to bottom: three frames of the first real video.

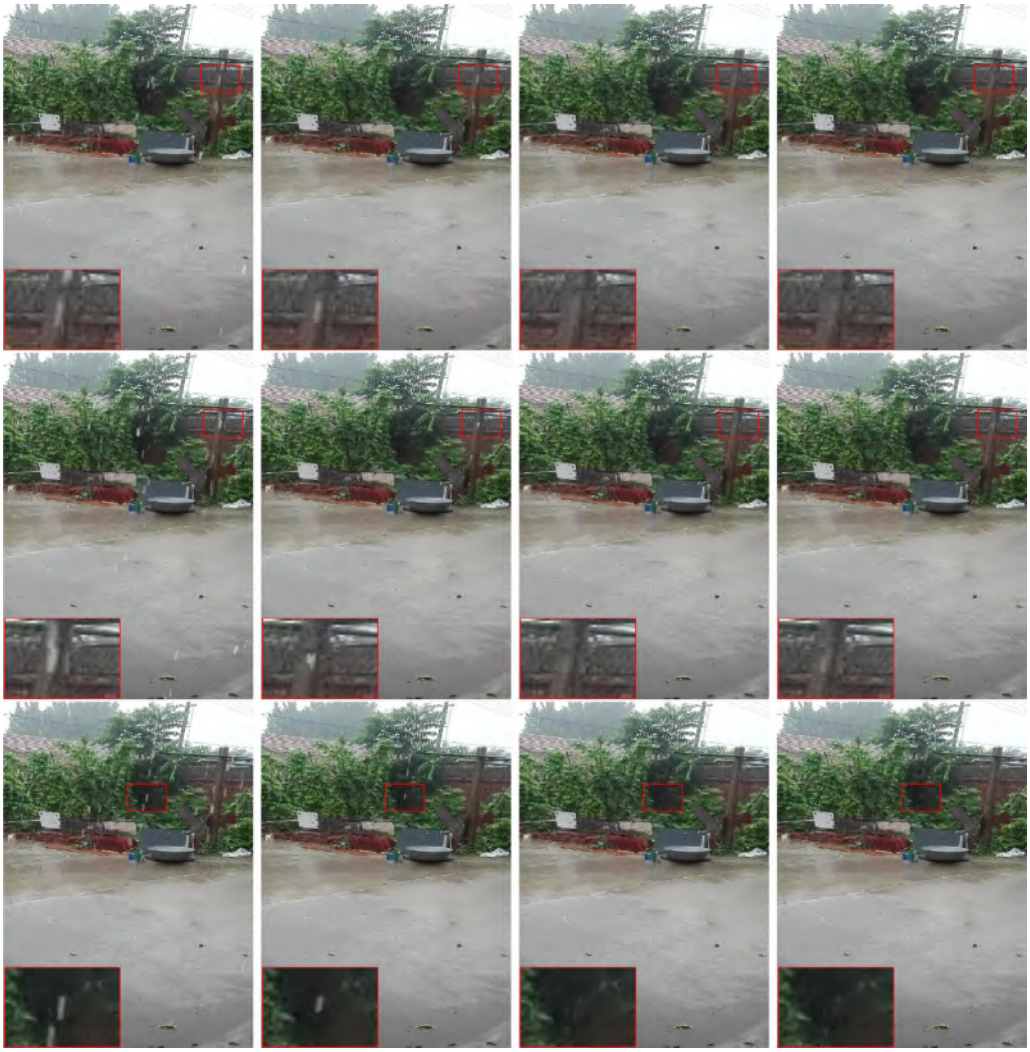

Fig. 11. Rain streak removal results of different methods on the video "yard". From left to right: the rainy frames, the results by LRMC [7], DIP [15], and the proposed method.

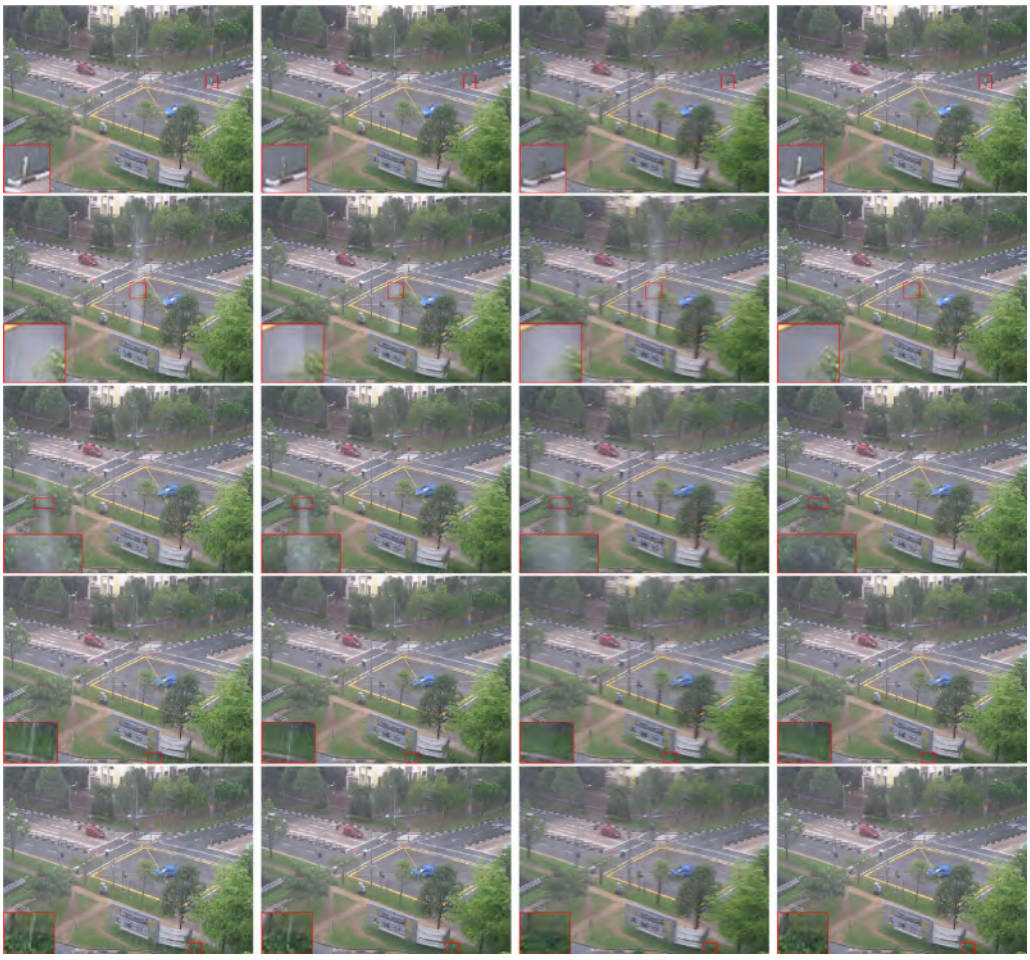

Fig. 12. Rain streak removal results of different methods on the video "crossing". From left to right: the rainy frames, the results by LRMC [7], DIP [15], and the proposed method.

is less than 5×10^{-3} or the iteration number is larger than 250. The results with the highest PSNR among the tested values are reported.

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

For visual comparisons, the frames of the estimated video are displayed in Figs. 6 and 7. It is observed that the proposed method achieves significantly better visual quality than the compared methods in terms of preserving details and removing rain streaks. There are two main reasons. On the one hand, LRMC and DIP both assume the clean video is low-rank, which leads to the loss of some obvious details. In the proposed method, we drop the low-rankness of the clean video. This choice helps the proposed method to handle the dynamic video better. For example, DIP and LRMC mistakenly remove the road signs in "highway" with both heavy rain streaks and light rain streaks. In "bridgefar", although the clean video is almost static, some details such as the ripples caused by the wind would destroy the low-rankness. DIP and LRMC fail to preserve ripples well. On the other hand, we use the group sparsity to characterize rain streaks, which helps to preserve the continuity of the rain streaks, leading to more accurate rain streak removal results than compared methods. In comparison, DIP fails to preserve the continuity of rain streaks, e.g., "coastguard" and "foreman" with heavy rain streaks and "carphone" with light rain streaks. Since the continuity of the rain streaks is significant for heavy rain streaks, it is easy to observe that the proposed method equipped with group sparsity term outperforms the compared methods in the videos with 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. Fig. 8 shows the PSNR curves of the proposed method using different parameter settings, where the testing parameter is chosen from the geometric series $\{0.1, 0.121, \ldots, 0.1\times1.1^k, \ldots, 1000\}$. It could be found that each component in the proposed method has an important contribution to the performance of the proposed method.

Discussion of groups: The group size is a vital parameter which is set as one column in this paper unless otherwise specified. Nevertheless, it is interesting to investigate the influence of 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 influence on the performance of the proposed model. The video with heavy rain streaks favors large group size while the video with light rain streaks favors small group size. For simplicity, we choose one column as default in all experiments because there is no significant difference among different group sizes.

Discussions of oblique rain streaks: Since we initially assumed that the rain streaks are vertical, the proposed method favors the case of vertical rain streaks. However, in practice, our assumption may not hold when the angle between the rain streaks and the vertical direction may be large. To test the performance of the proposed method on

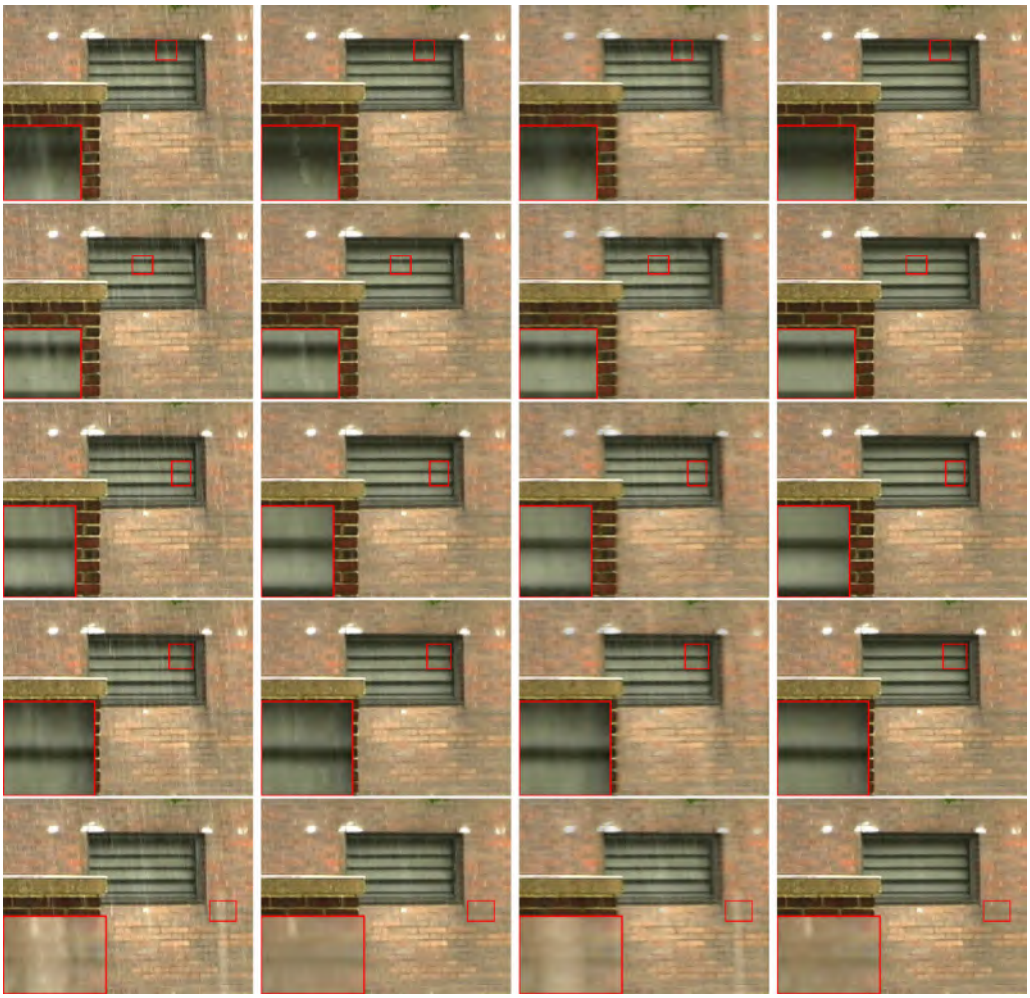

Fig. 13. Rain streak removal results of different methods on the video "wall". From left to right: the rainy frames, the results by LRMC [7], DIP [15], and the proposed method.

the videos with oblique rain streaks, we consider two synthetic videos "highway2" and "waterfall", where the angles between rain streaks and the vertical direction are randomly distributed in $[35°, 55°]$ in "highway2" and $[15°, 35°]$ in "waterfall", respectively. The quantitative results are given in Table 3 and the recovered frames are displayed in Fig. 9. We observe that the proposed method still obtains the best performance and achieves promising results. We give an explanation for this. The proposed model consists of four regularization terms, which simultaneously contribute to the rain streak removal. When the rain streaks are oblique, the regularizers corresponding to the directional property and the group sparsity of the rain streaks would not be helpful. Nonetheless, the temporal and horizontal continuity of the background still hold. Therefore, tuning the parameters to emphasize the effects of these two regularizers would help the proposed method to remove the rain streaks.

Discussions of the preprocessing: Before applying our algorithm, there are two preprocessing steps: (a) the conversion from RGB color space to YUV color space; (b) adding reflective boundary conditions. We 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. We observe that the preprocessing (a) does not have a significant influence on the performance but can significantly reduce the running time. Since LRMC is designed for RGB videos, we do the preprocessing (a) for DIP, while we still use the RGB format as the input of LRMC. On the other hand, as we expected, the reflective boundary condition slightly improves the performance. Based on the above discussions, we keep the two preprocessing steps in the proposed method for the performance and efficiency purpose.

4.2. Real data

We test four real rainy videos. The first one is a clipped part of size $260 \times 440 \times 3 \times 128$ from the movie "the Matrix", and the second video[4] (denoted as "yard") is a backyard video of size $512 \times 256 \times 3 \times 128$ recorded in a rainy day. The third video[5] (denoted as "crossing") of size $480 \times 640 \times 3 \times 108$ is captured in the crossing with complex traffic conditions, and the last video (denoted as "wall") of size $288 \times 368 \times 3 \times 173$ is download from the CAVE dataset[6]. 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: The first video is a very challenging video under lightning which enlarges the difference between adjacent frames and breaks the continuity along the temporal direction. The rain streak removal results are displayed in Fig. 10. We can observe that the rain streaks are more effectively removed by the proposed methods as compared with the other methods.

[4] https://github.com/TaiXiangJiang/FastDeRain/blob/master/yard.mp4.
[5] https://github.com/hotndy/SPAC-SupplementaryMaterials/blob/master/Dataset_Testing_RealRain/ra4_Rain.rar.
[6] http://www.cs.columbia.edu/CAVE/projects/camerarain/.

For the second video, the rain streak removal results are displayed in Fig. 11. The performance of DIP is promising when there is no moving foreground and dynamic background, which, however, is not always meet in the real-world. Nevertheless, the rain streaks are more effectively removed by the proposed methods as compared with the other methods.

The rain streak removal results for the third video and the last video are displayed in Figs. 12 and 13, respectively. We observe that the proposed method performs better than the competing methods. More precisely, with the help of the group sparsity, the proposed method removes rain streaks more completely than LRMC and DIP. On the other hand, by dropping the low-rankness regularizer, the proposed method preserves image details better than DIP. These results demonstrate the superiority of the proposed method again.

5. Conclusions

In this paper, we proposed a tensor-based rain streak removal model. We used the group sparsity and the smoothness along the vertical direction to characterize the rain streaks, and used the smoothness along the horizontal direction and the temporal direction to characterize the clean video. We developed 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 methods quantitatively and qualitatively.

Acknowledgments

The research is supported by NSFC, China (No. 61876203, 61772003, and 61702083), the Fundamental Research Funds for the Central Universities, China (No. ZYGX2016J132, ZYGX2016KYQD142, and ZYGX2016J129), and the National Postdoctoral Program for Innovative Talents, China (BX20180252).

References

[1] S. Maji, A.-C. Berg, J. Malik, Classification using intersection kernel support vector machines is efficient, in: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2008, pp. 1–8.
[2] O.-L. Junior, D. Delgado, V. Goncalves, U. Nunes, Trainable classifier-fusion schemes: An application to pedestrian detection, in: Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC), 2009, pp. 1–6.
[3] M.-S. Shehata, J. Cai, W.-M. Badawy, T.-W. Burr, M.-S. Pervez, R.-J. Johannesson, A. Radmanesh, Video-based automatic incident detection for smart roads: the outdoor environmental challenges regarding false alarms, IEEE Trans. Intell. Transp. Syst. 9 (2008) 349–360.
[4] X. Zhang, C. Zhu, S. Wang, Y.-P. Liu, M. Ye, A bayesian approach to camouflaged moving object detection, IEEE Trans. Circuits Syst. Video Technol. 27 (9) (2017) 2001–2013.
[5] C. Ma, Z.-J. Miao, X.-P. Zhang, M. Li, A saliency prior context model for real-time object tracking, IEEE Trans. Multimed. 19 (11) (2017) 2415–2424.
[6] F. Liu, Z.-G. Chen, J. Wang, Video image target monitoring based on rnn-lstm, Multimedia Tools Appl. (2018) http://dx.doi.org/10.1007/s11042-018-6058-6.
[7] J.-H. Kim, J.-Y. Sim, C.-S. Kim, Video deraining and desnowing using temporal correlation and low-rank matrix completion, IEEE Trans. Image Process. 24 (2015) 2658–2670.
[8] L.-W. Kang, C.-W. Lin, Y.-H. Fu, Automatic single-image-based rain streaks removal via image decomposition, IEEE Trans. Image Process. 21 (2012) 1742–1755.
[9] Y. Luo, Y. Xu, H. Ji, Removing rain from a single image via discriminative sparse coding, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 3397–3405.
[10] M. Roser, J. Kurz, 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), 2011, pp. 235–244.
[11] S.-D. You, R.-T. Tan, R. Kawakami, Y. Mukaigawa, K. Ikeuchi, Adherent raindrop modeling, detectionand removal in video, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2016) 1721–1733.
[12] L.-J. Deng, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, A directional global sparse model for single image rain removal, Appl. Math. Model. 59 (2018) 662–679.
[13] Y.-L. Chen, 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), 2013, pp. 1968–1975.
[14] Y. Li, R.-T. Tan, X.-J. Guo, J.-B. Lu, M.-S. Brown, Rain streak removal using layer priors, in: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2736–2744.
[15] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, 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), 2017, pp. 4057–4066.
[16] S.-H. Sun, S.-P. Fan, Y.-C.-F. Wang, Exploiting image structural similarity for single image rain removal, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 2014, pp. 4482–4486.
[17] L. Zhu, C.-W. Fu, D. Lischinski, P.-A. Heng, Joint bi-layer optimization for single-image rain streak removal, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2545–2553.
[18] Y. Chang, L.-X. Yan, S. Zhong, Transformed low-rank model for line pattern noise removal, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 1735–1743.
[19] D. Eigen, D. Krishnan, R. Fergus, Restoring an image taken through a window covered with dirt or rain, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2013, pp. 633–640.
[20] W.-H. Yang, R.-T. Tan, J.-S. Feng, J.-Y. Liu, Z.-M. Guo, 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), 2017, pp. 1685–1694.
[21] X.-Y. Fu, J.-B. Huang, X.-H. Ding, Y.-H. Liao, J. Paisley, Clearing the skies: A deep network architecture for single-image rain removal, IEEE Trans. Image Process. 26 (2017) 2944–2956.
[22] X.-Y. Fu, J.-B. Huang, D.-L. Zeng, Y. Huang, X.-H. Ding, J. Paisley, Removing rain from single images via a deep detail network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1715–1723.
[23] H. Zhang, V. Sindagi, V.-M. Patel, Image de-raining using a conditional generative adversarial network. arXiv preprint arXiv:1701.05957, 2017.
[24] R. Qian, R.-T. Tan, W.-H. Yang, J.-J. Su, J.-Y. Liu, Attentive generative adversarial network for raindrop removal from a single image, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 2482–2491.
[25] S.-Y. Li, W.-Q. Ren, J.-W. Zhang, J.-K. Yu, X.-J. Guo, Fast single image rain removal via a deep decomposition-composition network. arXiv preprint arXiv:1804.02688, 2018.
[26] H. Zhang, V.-M. Patel, Density-aware single image de-raining using a multi-stream dense network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 695–704.
[27] W. Wei, D.-Y. Meng, Q. Zhao, Z.-B. Xu, Semi-supervised CNN for single image rain removal. arXiv preprint arXiv:1807.11078, 2018.
[28] K. Garg, S.-K. Nayar, Detection and removal of rain from videos, in: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2004, pp. 528–535.
[29] A.-K. Tripathi, S. Mukhopadhyay, Video post processing: low-latency spatiotemporal approach for detection and removal of rain, IET Image Process. 6 (2012) 181–196.
[30] W. Wei, L.-X. Yi, Q. Xie, Q. Zhao, D.-Y. Meng, 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), 2017, pp. 2535–2544.
[31] J. Chen, C.-H. Tan, J.-H. Hou, L.-P. Chau, H. Li, Robust video content alignment and compensation for rain removal in a cnn framework, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 6286–6295.
[32] J.-Y. Liu, W.-H. Yang, S. Yang, Z.-M. Guo, Erase or fill? deep joint recurrent rain removal and reconstruction in videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 3233–3242.
[33] A.-K. Tripathi, S. Mukhopadhyay, Removal of rain from videos: a review, Signal Image Video Process. 8 (2014) 1421–1430.
[34] Y.-P. Liu, Z. Long, C. Zhu, Image completion using low tensor tree rank and total variation minimization, IEEE Trans. Multimed. (2018) http://dx.doi.org/10.1109/TMM.2018.2859026.
[35] M.-Y. Zhou, Y.-P. Liu, Z. Long, L.-X. Chen, C. Zhu, Tensor rank learning in cp decomposition via convolutional neural network, Signal Processing Image Communication (2018) http://dx.doi.org/10.1016/j.image.2018.03.017.
[36] T.-Y. Ji, T.-Z. Huang, X.-L. Zhao, T.-H. Ma, L.-J. Deng, A non-convex tensor rank approximation for tensor completion, Appl. Math. Model. 48 (2017) 410–422.
[37] X.-L. Zhao, W. Wang, T.-Y. Zeng, T.-Z. Huang, M.-K. Ng, Total variation structured total least squares method for image restoration, SIAM J. Sci. Comput. 35 (2013) 1304–1320.
[38] X.-L. Zhao, F. Wang, M.-K. Ng, A new convex optimization model for multiplicative noise and blur removal, SIAM J. Imaging Sci. 7 (2014) 456–475.
[39] L.-J. Deng, W.-H. Guo, T.-Z. Huang, Single image super-resolution by approximated heaviside functions, Inform. Sci. 348 (2016) 107–123.
[40] L.-J. Deng, W.-H. Guo, T.-Z. Huang, Single-image super-resolution via an iterative reproducing kernel hilbert space method, IEEE Trans. Circuits Syst. Video Technol. 26 (2016) 2001–2014.
[41] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, T.-Y. Ji, L.-J. Deng, Matrix factorization for low-rank tensor completion using framelet prior, Inform. Sci. 436 (2018) 403–417.

[42] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, Y. Wang, FastDeRain: A novel video rain streak removal method using directional gradient priors, IEEE Trans. Image Process. (2018) http://dx.doi.org/10.1109/TIP.2018.2880512.

[43] L.-B. Cui, C. Chen, W. Li, M.-K. Ng, An eigenvalue problem for even order tensors with its applications, Linear Multilinear Algebra 64 (2016) 602–621.

[44] W. Deng, W.-T. Yin, Y. Zhang, Group sparse optimization by alternating direction method, in: Proceedings of the The International Society of Optics and Photonics (SPIE), 2013.

[45] X.-J. Guo, Y. Ma, Generalized tensor total variation minimization for visual data recovery? in: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3603–3611.

[46] Y.-Q. Jiang, X. Jin, Z.-Y. Wu, Video inpainting based on joint gradient and noise minimization, in: Pacific-Rim Conference on Advances in Multimedia Information Processing, 2016, pp. 407–417.

[47] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (2011) 1–122.

[48] Z. Wang, A.-C. Bovik, H.-R. Sheikh, E.-P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (2004) 600–612.