

RoDeRain: Rotational Video Derain via Nonconvex and Nonsmooth Optimization

Lizhen Deng¹ · Guoxia Xu² · Hu Zhu³ ·
Bing-Kun Bao⁴

Received: date / Accepted: date

Abstract Video derain is an important issue in the field of digital image processing and computer vision. This paper divides rain streaks into two types: one is rain in natural scenes, and the other is rain in stochastic scenes. In this paper, we propose a novel rotational video derain algorithm via nonconvex and nonsmooth algorithm (RoDerain). Not only can the rain streaks in natural scene be removed, but the rain streaks in stochastic scene can be also well removed. This paper added the rotation operator based on the discriminatively intrinsic priors of rain streaks and clean videos to remove the rain streaks in both natural and stochastic scenes. For the low rank problem of the background, we replace the solution of the nuclear norm with improved IRNN-Capped L_1 suitable for tensor. Finally, this paper used the Alternating Direction Method of Multipliers (ADMM) to optimize the solution of the proposed rain streaks removal algorithm model. The disadvantage is that global information is not considered. And the extensive experiment results show that our proposed algorithm performs favorably in comparison to several popular rain removal algorithms.

corresponding author: Bin-Kun Bao
E-mail: bingkunbao@njupt.edu.cn

¹ National Engineering Research Center of Communication and Network Technology, Nanjing University of Posts and Telecommunications, Nanjing, 210003, People's Republic of China

² Department of Computer Science, Norwegian University of Science and Technology, 2815 Gjøvik, Norway

³ Jiangsu Province Key Lab on Image Processing and Image Communication, Nanjing University of Posts and Telecommunications, Nanjing, 210003, People's Republic of China

⁴ College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, 210003, People's Republic of China

Keywords Rain Removal · Rotation Operator · Low Rank

1 Introduction

With the advancement of computer technology, the outdoor vision system is widely used in target recognition [9] and tracking [5], traffic monitoring [29], remote sensing monitoring [22] and military reconnaissance [28], etc. Outdoor vision systems are often affected by various weather conditions, especially rain, snow, fog and haze. These bad weather conditions will lead to serious degradation of the image or video captured by the outdoor vision system, which brings difficulty in subsequent video processing tasks. Therefore, processing these degraded images and videos is of great significance.

The rain removal method began to emerge from time-domain filtering method proposed by Starik et al. [19] in 2003. Although it started late, the rain technology has become a hot spot for scholars at home and abroad in recent years. There have been many methods of rain streaks removal based on sparse domains. We mainly introduce the rain streaks removal algorithms from two aspects of matrix and tensor.

For the matrix-based method. Kim et al. [11] used the time-spatial continuity, and combined the support vector machine (SVM) [3] and low-rank matrix completion [20] [2] [18] to reach the rain streaks. And this kind of method is time-consuming. Manu et al. [1] effectively uses the L_0 gradient minimization approach to remove the rain pixels. Li et al. [25] proposed an effective method that uses simple patch-based priors for both the background and rain layers. These priors are based on Gaussian mixture models and can accommodate multiple orientations and scales of the rain streaks. Zhu et al. [17] presented a method for removing rain streaks from a single input image by decomposing it into a rain-free background layer and a rain-streak layer.

For the tensor-based method. Chen et al. [24] proposed the rain removal model of low-rank from matrix to tensor structure for the detection of time-space interrelated rain streaks. ALAA [7] considered that rain-free scenes are low rank for adjacent frames, so it can remove rain by extracting low rank components between adjacent frames. This kind of method has weak constraint and may have false detection of rain streaks. Jiang et al. [21] proposed a novel tensor based video rain streaks removal approach by fully considering the discriminatively intrinsic characteristics of rain streaks and clean videos.

Removing rain streaks from video has important practical implications. For example, for a safe driving assistance system in an intelligent transportation application, the removal of rain streaks in the video can improve the visibility of the rainy day, which enable the driver to drive more comfortably and clearly, and reduce the incidence of traffic accidents.

In recent years, deep learning-based methods are very popular. Zeng et al. [27] proposed a lightweight channel spatial attention network for real-time image de-raining. YU et al. [26] proposed a progressive network for single-

image rain removal. By removing rain bars of various densities, proportions and shapes, the image details of the rain-free area are well preserved.

However, there are still limitations existing in current rain removal methods. Most existing rain streaks removal algorithms are effective for the rain in natural scenes, but it is different to remove the rain streaks in 'stochastic' scenes. The difference between natural scenes and stochastic scenes is whether rain streaks trajectory is perpendicular to the horizontal direction. The traditional preprocessing step is that the image will be rotated manually, but it is intractable for stochastic scenes because of the complexity of algorithm and unpredictable parameter of rotational angle. This paper adds the rotation operator to the existing rain streaks removal model based on the discriminative characteristics of rain streaks and the clean video in the gradient domain to solve the rain streaks in both natural and stochastic scenes.

The low-rank minimization problem is NP-hard. Most current algorithms use nuclear norm regularization to solve this low-rank problem. For example, Jiang et.al [21] considered the low-rank problem of tensor as a convex optimization problem, and used nuclear norm of tensor to solve. This paper considered the low-rank minimization problem of clean video as a non-convex non-smooth sparse minimization problem, and generalized the Iteratively Reweighted Nuclear Norm (IRNN) of the matrix to the tensor, and combined the nuclear norm of tensor with Capped L_1 . This allows detailed processing of background information and improve the quality of the restored image.

This paper used the Alternating Direction Method of Multipliers (ADMM) to optimize the solution of the proposed rain streaks removal algorithm model. Based on the idea of divide and conquer, ADMM divides a large overall problem into multiple local sub-problems, which reduces the scale of the problem. It is especially suitable for machine learning problems.

Based the above aspects, we propose an adaptive low-rank tensor completion derain algorithm. The specific algorithm will be explained in detail in the following sections.

The paper is organized as follows. In Section 2, we introduce the basic tensor notations, which made a good preparation for the following. Section 3 mainly introduces the construction of the Adaption low-rank tensor completion for derain Algorithm. The whole detailed optimization processing of the proposed rain streaks algorithm is presented in Section 4. In Section 5, real-world experimental results, simulated experimental results and discussion are presented. Finally, Section 6 is the conclusion of this paper.

2 Tensor notations

The existing knowledge of vector and matrix space has been unable to deal with the problem of video, which has prompted people to learn and use tensor-related knowledge for in-depth research, which has attracted extensive attention from experts and scholars at home and abroad.

In general, we use lowercase letters to represent vectors, uppercase letters to represent matrices, and swashes to represent tensors. For example, x is a vector, X is a matrix, \mathcal{X} is a tensor. The scalar can be represented by a zero-order tensor, the vector can be represented by a first-order tensor, and the matrix can be represented by a second-order tensor. The tensor can be interpreted as a high-order generalization of vectors and matrices (orders greater than or equal to three), which can better represent high-order data with complex structures. An N -order tensor \mathcal{A} can be defined as $\mathcal{A} \in \mathcal{R}^{I_1 \times I_2 \times \dots \times I_N}$, and its (i_1, i_2, \dots, i_N) -th element is $a_{i_1 i_2 \dots i_N}$. The following is a brief introduction to the basic operation of tensor.

Definition 1: (The inner product of tensor) Two N -order tensors \mathcal{A} and \mathcal{B} have the same size ($\mathcal{A}, \mathcal{B} \in \mathcal{R}^{I_1 \times I_2 \times \dots \times I_N}$), then their inner product can be defined as:

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1, i_2, \dots, i_N} a_{i_1 i_2 \dots i_N} \cdot b_{i_1 i_2 \dots i_N} \quad (1)$$

Definition 2: (The Frobenius norm of tensor) N -order tensor \mathcal{A} 's Frobenius norm is the square root of all its elements:

$$\|\mathcal{A}\|_F = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle} = \left(\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} |a_{i_1 i_2 \dots i_N}|^2 \right)^{1/2} \quad (2)$$

The L_1 norm of tensor: N -order tensor \mathcal{A} 's L_1 norm is the sum of all non-zero elements.

The L_0 norm of tensor: N -order tensor \mathcal{A} 's L_0 norm is the number of all non-zero elements.

In this paper, the order of the tensor is three. For example, is third-order tensor. Taking the third-order tensor $\mathcal{A} \in \mathcal{R}^{I \times J \times K}$ as an example, it has three kinds of slices. There are horizontal slice $\mathcal{A}_{i,:}$, lateral slice $\mathcal{A}_{:,j}$, and frontal slice $\mathcal{A}_{:,k}$. The mode- unfolding of third-order tensor is the frontal slice $\mathcal{A}_{:,k}$, the horizontal slice $\mathcal{A}_{i,:}$, and the lateral slice $\mathcal{A}_{:,j}$ is devised according to the horizontal direction respectively.

3 The prerequisites of video image derain

The observation model [15] [14] [23] of rainy video can be formulated as :

$$\mathcal{O} = \mathcal{B} + \mathcal{R} \quad (3)$$

where \mathcal{O} , \mathcal{B} and \mathcal{R} represent the rainy video, the unknown rain-free video and rain streaks, respectively. They are all three tensors of size $m \times n \times t$.

Clean rain-free video \mathcal{B} and the rain \mathcal{R} can be recovered from rainy video \mathcal{O} , which is an ill-posed problem. Thus, this requires us to solve this inverse problem through prior knowledge. There are three important prior knowledge.

3.1 Rain Streaks Removal Model Based on Tensor

In 2017, Jiang et al. [21] utilized Discriminatively Intrinsic Priors to removal the rain streaks. Firstly, Rain streaks are sparse and smooth along the direction of raindrop. And then, the clean videos possess smoothness along the rain-perpendicular direction and global and local correlation along time direction. At last, the clean rain-free videos have the characteristic of low-rank because of the little change of background. Thus, they proposed the rain removal model via Discriminatively Intrinsic Priors:

$$\begin{aligned} \min_{\mathcal{R}} \quad & \alpha_1 \|\nabla_y \mathcal{R}\|_1 + \alpha_2 \|\mathcal{R}\|_1 + \|\mathcal{O} - \mathcal{R}\|_* \\ & + \alpha_3 \|\nabla_x(\mathcal{O} - \mathcal{R})\|_1 + \alpha_4 \|\nabla_t(\mathcal{O} - \mathcal{R})\|_1 \end{aligned} \quad (4)$$

where ∇_y and ∇_x are difference operators along vertical and horizontal directions, respectively, and ∇_t indicates the time directional difference operator. $\|\cdot\|_*$ indicates the nuclear norm.

Fig. 1 shows some of the statistics characteristics of rain streaks and rain-free video. Firstly, the first row shows the rainy frame, clean frame and rain streaks respectively. And then, (a-1,2,3) are the histograms of the rainy frame, clean frame and the rain streaks respectively. (b-1,2,3) are the histograms of from the rainy frame, clean frame and the rain streaks. We can observe that (a-3) has the most zero values among (a-1,2,3), and (b-3) has the most zero values among (b-1,2,3), thus the raindrops are sparse and the raindrops are also sparse along the direction of the drop. And then, (c-1,2,3) are the histograms of difference along the horizontal direction from the rainy frame, clean frame and the rain streaks respectively. We can see that many burrs appear in (c-1) and (c-3), while (c-2) is relatively piecewise smooth. The clean frame is smooth along the horizontal direction. And so on, the clean video is smooth along the horizontal direction. At last, (d-1,2,3) are the intensity value curve of a fixed pixel of every frame from the rainy video, clean video and rain streaks. And we find that (d-2) has the smoothest change.

3.2 Iteratively Reweighted Nuclear Norm

To solve the matrix problem of low rank, which is NP-hard, the measure of nuclear norm may not get the perfect solution. Thus, Lu et al. [4] proposed a non-convex non-smooth low-rank minimization method called Iteratively Reweighted Nuclear Norm (IRNN) to solve this kind of problem about matrix.

$$\min \lambda \text{rank}(X) + \frac{1}{2} \|X - Y\|_F^2 \quad (5)$$

where X and Y is the matrix of size of $m \times n$, $\lambda > 0$, $0 \leq \omega_1 \leq \omega_2 \leq \dots \leq \omega_s$ and $s = \min(m, n)$.

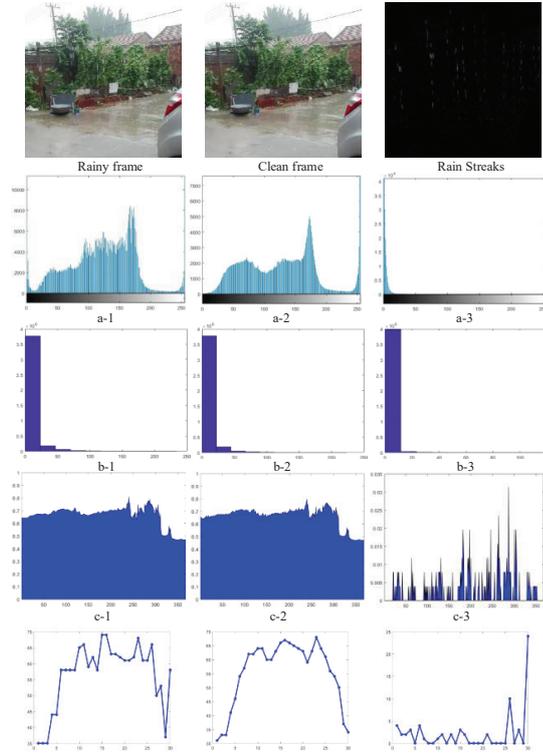


Fig. 1 (a-1,2,3) are the histograms of the rainy frame, clean frame and the rain streaks respectively; (b-1,2,3) are the histograms of difference along the vertical direction from the rainy frame, clean frame and the rain streaks respectively; (c-1,2,3) are the histograms of difference along the horizontal direction from the rainy frame, clean frame and the rain streaks respectively; (d-1,2,3) are the intensity value curve of a fixed pixel of every frame from the rainy video, clean video and rain streaks.

The solution of Eq.(5) weights the singular values of the matrix, the Eq.(5) can be transformed as:

$$\min \lambda \sum_{i=1}^s \omega_i \sigma_i(X) + \frac{1}{2} \|X - Y\|_F^2 \quad (6)$$

where $\sigma_i(X)$ is the i -th singular value of X . Thus, X can be solved by:

$$X^* = US_{\lambda\omega}(\Sigma)V^T \quad (7)$$

where $Y = U\Sigma V^T$ is the singular value of Y , and $S_{\lambda\omega}(\Sigma) = \text{Diag}\{(\Sigma_{ii} - \lambda\omega_i)\}$.

4 Rotational video derain via nonconvex and nonsmooth optimization

4.1 The motivation of the proposed model

4.1.1 The Structural Element of Rotation Operator

Affected by various external influences, the trajectory of rain streaks is not always perpendicular to the ground. There is a certain inclination angle between the trajectory of rain streaks and ground. In this paper, we refer to this kind of rain streaks as the rain streaks in stochastic scenes. And at the same time, we refer to the rain streaks whose trajectory is perpendicular to the horizontal direction as the rain streaks in natural scenes. However, for the removal processing of the rain streaks in stochastic scenes, many rain streaks removal algorithms need to rotate every frame of the rainy video. If the length of the video is very long, that is, the number of frames of the video is very large, it can be very cumbersome. So this paper added the rotation operator \mathcal{A}_θ to Fig. 4's the first item and fourth item. $\mathcal{A}_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$ and $\mathcal{A}_\theta^T = \mathcal{A}_{-\theta}$.

The first item of the original model changed from $\alpha_1 \|\nabla_y \mathcal{R}\|_1$ to $\alpha_1 \|\nabla_y (\mathcal{A}_\theta \mathcal{R})\|_0$, and The fourth item of the original model changed from $\alpha_3 \|\nabla_x (\mathcal{O} - \mathcal{R})\|_1$ to $\alpha_3 \|\nabla_x (\mathcal{A}_\theta (\mathcal{O} - \mathcal{R}))\|_0$. The combination of the rotation operator and the L_0 norm can better correct and remove rain streaks in stochastic scenes. Subsequent experiment results can also better confirm the effectiveness of the rotation operator.

4.1.2 IRNN - Capped L_1

In this paper, to solve the low rank problem of tensor, we extend IRNN from solving matrix problem to tensor problem. And the Eq.(3) can be transformed as:

$$\min \lambda \frac{1}{3} \sum_{n=1}^3 \sum_{i=1}^3 \omega_i \sigma_i(\mathcal{X}_{(n)}) + \frac{1}{2} \|\mathcal{X} - \mathcal{Y}\|_F^2 \quad (8)$$

where $\mathcal{X}_{(n)} = \text{Unfold}_n(\mathcal{X})$, $\mathcal{X}, \mathcal{Y} \in \mathcal{R}^{m \times n \times t}$. It is more suitable for our algorithm model. λ is a weight. In this paper, the order of the tensor is three.

Let $P_i = \text{Unfold}_i(\mathcal{Y})$ ($i = 1, 2, 3$), suppose the singular value decomposition of P_i is $P_i = U \Sigma V^T$, and $Q_i = U S_{\lambda \omega}(\Sigma) V^T$, thus the optimal solution \mathcal{X} is :

$$\mathcal{X}^* = \sum_{i=1}^3 \frac{1}{3} \text{Fold}_i(Q_i) \quad (9)$$

where $S_{\lambda \omega}(\Sigma) = \text{Diag}\{(\Sigma_{ii} - \lambda \omega_i)\}$, $\text{Fold}_i(Q_i)$ is to fold Q_i to a tensor with the same size of \mathcal{X} . The value of ω_i is selected by the super-gradient of Capped L_1 [4] as shown in Fig.2.

4.2 The proposed Derain Model

In this paper, we add the rotation operator based on the discriminatively intrinsic priors of rain streaks and clean videos to remove the rain streaks in both natural and stochastic scenes. For the low rank problem of the background, we replace the solution of the nuclear norm with improved IRNN-Capped L_1 suitable for tensor. Based on the above, the final proposed rain streaks algorithm model via adaptive low-rank tensor completion in this paper is:

$$\begin{aligned} \min_{\mathcal{R}} \quad & \alpha_1 \|\nabla_y(\mathcal{A}_\theta \mathcal{R})\|_0 + \alpha_2 \|\mathcal{R}\|_1 + \alpha_3 \|\nabla_x(\mathcal{A}_\theta(\mathcal{O} - \mathcal{R}))\|_0 \\ & + \alpha_4 \|\nabla_t(\mathcal{O} - \mathcal{R})\|_1 + \|\mathcal{O} - \mathcal{R}\|_{Capped L1} \end{aligned} \quad (10)$$

4.3 Alternating direction multipliers optimization

In this section, it mainly introduces the mainly optimization processing of our algorithm model.

To the first, we need to introduce five auxiliary tensors \mathcal{Y} , \mathcal{S} , \mathcal{X} , \mathcal{T} and \mathcal{L} , and form such the constraints of $\nabla_y(\mathcal{A}_\theta \mathcal{R}) = \mathcal{Y}$, $\mathcal{R} = \mathcal{S}$, $\nabla_x(\mathcal{A}_\theta(\mathcal{O} - \mathcal{R})) = \mathcal{X}$, $\nabla_t(\mathcal{O} - \mathcal{R}) = \mathcal{T}$ and $\mathcal{O} - \mathcal{R} = \mathcal{L}$ to solve the proposed optimization problem by Alternating Direction Method of Multipliers (ADMM) [8]. So, our algorithm model is transformed as:

$$\begin{aligned} \min_{\mathcal{R}, \mathcal{Y}, \mathcal{S}, \mathcal{X}, \mathcal{T}, \mathcal{L}} \quad & \alpha_1 \|\mathcal{Y}\|_0 + \alpha_2 \|\mathcal{S}\|_1 + \alpha_3 \|\mathcal{X}\|_0 + \alpha_4 \|\mathcal{T}\|_1 + \|\mathcal{L}\|_{Capped L1} \\ s.t. \quad & \nabla_y(\mathcal{A}_\theta \mathcal{R}) = \mathcal{Y}, \quad \mathcal{R} = \mathcal{S}, \quad \nabla_x(\mathcal{A}_\theta(\mathcal{O} - \mathcal{R})) = \mathcal{X}, \\ & \nabla_t(\mathcal{O} - \mathcal{R}) = \mathcal{T}, \quad \mathcal{O} - \mathcal{R} = \mathcal{L} \end{aligned} \quad (11)$$

where $\mathcal{Y}, \mathcal{S}, \mathcal{X}, \mathcal{T}$ and $\mathcal{L} \in \mathcal{R}^{m \times n \times t}$. Thus, the augmented Lagrangian function is:

$$\begin{aligned} L(\mathcal{R}, \mathcal{Y}, \mathcal{S}, \mathcal{T}, \mathcal{L}, \Lambda) = & \alpha_1 \|\mathcal{Y}\|_0 + \alpha_2 \|\mathcal{S}\|_1 + \alpha_3 \|\mathcal{X}\|_0 + \alpha_4 \|\mathcal{T}\|_1 + \|\mathcal{L}\|_{Capped L1} \\ & + \langle \Lambda_1, \nabla_y(\mathcal{A}_\theta \mathcal{R}) - \mathcal{Y} \rangle + \frac{\beta_1}{2} \|\nabla_y(\mathcal{A}_\theta \mathcal{R}) - \mathcal{Y}\|_2^2 \\ & + \langle \Lambda_2, \mathcal{R} - \mathcal{S} \rangle + \frac{\beta_2}{2} \|\mathcal{R} - \mathcal{S}\|_2^2 \\ & + \langle \Lambda_3, \nabla_x(\mathcal{A}_\theta(\mathcal{O} - \mathcal{R})) - \mathcal{X} \rangle + \frac{\beta_3}{2} \|\nabla_x(\mathcal{A}_\theta(\mathcal{O} - \mathcal{R})) - \mathcal{X}\|_2^2 \\ & + \langle \Lambda_4, \nabla_t(\mathcal{O} - \mathcal{R}) - \mathcal{T} \rangle + \frac{\beta_4}{2} \|\nabla_t(\mathcal{O} - \mathcal{R}) - \mathcal{T}\|_2^2 \\ & + \langle \Lambda_5, (\mathcal{O} - \mathcal{R}) - \mathcal{L} \rangle + \frac{\beta_5}{2} \|\mathcal{O} - \mathcal{R} - \mathcal{L}\|_2^2 \end{aligned} \quad (12)$$

where $\Lambda = [\Lambda_1, \Lambda_2, \dots, \Lambda_5]$ is the Multipliers of the augmented Lagrangian function and $\beta = [\beta_1, \beta_2, \dots, \beta_5]$ are five positive parameters.

The solution of the algorithm model is transformed from a constrained problem to an unconstrained one.

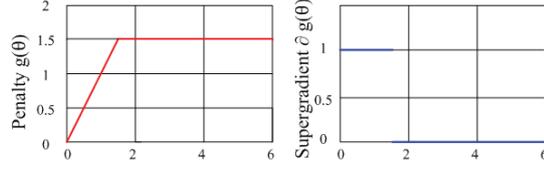


Fig. 2 Capped L_1 Penalty

– The sub-problems of \mathcal{S} , \mathcal{T} :

The first step is to solve the L_1 norm problem in the algorithm model about \mathcal{S} , \mathcal{T} . The sub-problem of \mathcal{S} is

$$\min_{\mathcal{S}} \alpha_2 \|\mathcal{S}\|_1 + \frac{\beta_2}{2} \left\| \mathcal{S} - \left(\mathcal{R} + \frac{\Lambda_2}{\beta_2} \right) \right\|_F^2 \quad (13)$$

It has a close-form solution by soft thresholding [6] as follows:

$$\mathcal{S}^{k+1} = \text{Shrink}_{\frac{\alpha_2}{\beta_2}} \left(\mathcal{R}^k + \frac{\Lambda_2^k}{\beta_2} \right) \quad (14)$$

At the same time, the \mathcal{T} -related sub-problem also belong to the problem of L_1 norm, and has the similar solution as follows:

$$\begin{aligned} \mathcal{T}^{k+1} &= \arg \min_{\mathcal{T}} \alpha_4 \|\mathcal{T}\|_1 + \frac{\beta_4}{2} \left\| \mathcal{T} - \left(\nabla_t(\mathcal{O} - \mathcal{R}) + \frac{\Lambda_4}{\beta_4} \right) \right\|_F^2 \\ &= \text{Shrink}_{\frac{\alpha_4}{\beta_4}} \left(\nabla_t(\mathcal{O} - \mathcal{R}^k) + \frac{\Lambda_4^k}{\beta_4} \right) \end{aligned} \quad (15)$$

– The sub-problems of \mathcal{X} , \mathcal{Y} :

Subsequently, the sub-problems of \mathcal{X} and \mathcal{Y} are both about the solution with L_0 norm. In [13], these problems can be solved by regularizing the sparsity property. Thus, the sub-problems of \mathcal{X} and \mathcal{Y} can be update respectively by:

$$\begin{aligned} \mathcal{Y}^{k+1} &= \arg \min_{\mathcal{Y}} \beta_1 \|\mathcal{Y}\|_0 + \left\| \mathcal{Y} - \left(\nabla_y(\mathcal{A}_\theta \mathcal{R}) + \frac{\Lambda_1}{\beta_1} \right) \right\|_2^F \\ &= \begin{cases} \nabla_y(\mathcal{A}_\theta \mathcal{R}^k) + \frac{\Lambda_1^k}{\beta_1}, & \left| \nabla_y(\mathcal{A}_\theta \mathcal{R}^k) + \frac{\Lambda_1^k}{\beta_1} \right|^2 \geq \beta_1 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (16)$$

$$\begin{aligned} \mathcal{X}^k &= \arg \min_{\mathcal{X}} \beta_3 \|\mathcal{X}\|_0 + \left\| \mathcal{X} - \left(\nabla_x(\mathcal{A}_\theta(\mathcal{O} - \mathcal{R}^k)) + \frac{\Lambda_3}{\beta_3} \right) \right\|_2^F \\ &= \begin{cases} \nabla_x(\mathcal{A}_\theta(\mathcal{O} - \mathcal{R}^k)) + \frac{\Lambda_3^k}{\beta_3}, & \left| \nabla_x(\mathcal{A}_\theta(\mathcal{O} - \mathcal{R}^k)) + \frac{\Lambda_3^k}{\beta_3} \right|^2 \geq \beta_3 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (17)$$

– The sub-problems of \mathcal{L} :

The sub-problems of \mathcal{L} can be solved by

$$\arg \min_{\mathcal{L}} \|\mathcal{L}\|_{Capped L_1} + \frac{\beta_5}{2} \left\| \mathcal{O} - \mathcal{R} - \mathcal{L} + \frac{\Lambda_5}{\beta_5} \right\|_F^2 \quad (18)$$

It is an important innovation of this paper. According the subsection named Iteratively Reweighted Nuclear Norm-Capped L_1 of Tensor of Section V, we can get the solution of \mathcal{L} is:

$$\mathcal{L}^{k+1} = \sum_{i=1}^3 \frac{1}{3} \text{Fold}_i(Q_i^k) \quad (19)$$

where $P_i = \text{Unfold}_i(\mathcal{O} - \mathcal{R} - \frac{\Lambda_3}{\beta_3}) = U\Sigma V^T$ and $Q_i = US_{\lambda\omega}(\Sigma)V^T$.

– The sub-problems of \mathcal{R} :

It is a least squares problems as follows:

$$\begin{aligned} \min_{\mathcal{R}} & \frac{\beta_1}{2} \left\| \nabla_y(\mathcal{A}_\theta \mathcal{R}) - \mathcal{Y} + \frac{\Lambda_1}{\beta_1} \right\|_F^2 + \frac{\beta_2}{2} \left\| \mathcal{R} - \mathcal{S} + \frac{\Lambda_2}{\beta_2} \right\|_F^2 \\ & + \frac{\beta_3}{2} \left\| \nabla_x(\mathcal{A}_\theta(\mathcal{O} - \mathcal{R})) - \mathcal{T} + \frac{\Lambda_3}{\beta_3} \right\|_F^2 \\ & + \frac{\beta_4}{2} \left\| \nabla_t(\mathcal{O} - \mathcal{R}) - \mathcal{X} + \frac{\Lambda_4}{\beta_4} \right\|_F^2 + \frac{\beta_5}{2} \left\| \mathcal{O} - \mathcal{R} - \mathcal{L} + \frac{\Lambda_5}{\beta_5} \right\|_F^2 \end{aligned} \quad (20)$$

Because of the difference operators and rotation operator in the formulations, we use fast Fourier transform(FFT) and its inverse fast Fourier transform (IFFT) to solve the problem, which can speed up the solution of the problem. So, \mathcal{R} can be updated by:

$$\mathcal{R}^{k+1} = \mathcal{F}^{-1} \left(\frac{\mathcal{F}(\mathbf{K}_1)}{\mathcal{F}(\mathbf{K}_2)} \right) \quad (21)$$

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

$$\begin{aligned} \mathbf{K}_1 &= \nabla_y^T \mathbf{A}_{-\theta} (\beta_1 \mathbf{Y}^k - \Lambda_1^k) + \nabla_x^T \mathbf{A}_{-\theta} (\beta_3 \nabla_x \mathbf{A}_\theta \mathbf{O} - \beta_3 \mathbf{T}^k) + \beta_5 (\mathbf{O} - \mathbf{L}^k) \\ &+ \Lambda_5^k + \nabla_t^T (\beta_4 \nabla_t \mathbf{O} - \beta_4 \mathbf{X}^k + \Lambda_4^k) - \Lambda_2^k + \beta_2 \mathbf{S}^k \\ \mathbf{K}_2 &= \beta_1 \nabla_y^T \nabla_y + (\beta_2 + \beta_5) \mathbf{I} + \beta_3 \nabla_x^T \nabla_x + \beta_4 \nabla_t^T \nabla_t \end{aligned}$$

Finally, these multipliers of ADMM can be updated parallel as follows:

$$\begin{aligned} \Lambda_1^{k+1} &= \Lambda_1^k + \beta_1 (\nabla_y(\mathcal{A}_\theta \mathcal{R}^k) - \mathcal{Y}^k) \\ \Lambda_2^{k+1} &= \Lambda_2^k + \beta_2 (\mathcal{R}^k - \mathcal{S}^k) \\ \Lambda_3^{k+1} &= \Lambda_3^k + \beta_3 (\nabla_x(\mathcal{A}_\theta(\mathcal{O} - \mathcal{R}^k)) - \mathcal{T}^k) \\ \Lambda_4^{k+1} &= \Lambda_4^k + \beta_4 (\nabla_t(\mathcal{O} - \mathcal{R}^k) - \mathcal{X}^k) \\ \Lambda_5^{k+1} &= \Lambda_5^k + \beta_5 (\mathcal{O} - \mathcal{R}^k - \mathcal{L}^k) \end{aligned} \quad (22)$$

Algorithm 1 RoDerain**Input:** The rainy video \mathcal{O} , θ **While not converge do**step1: Update \mathcal{S}, \mathcal{T} respectively by Eq.(13)-Eq.(15);step2: Update \mathcal{Y}, \mathcal{X} respectively by Eq.(16)-Eq.(17);step3: Update \mathcal{L} by Eq.(19);step4: Update \mathcal{R} by Eq.(21);

step5: Update the multipliers by Eq.(22);

end while**Output:** The estimation of rain-free video $\mathcal{B} = \mathcal{O} - \mathcal{R}$ and rain streaks \mathcal{R} .

In summary, the algorithm of ADMM decomposes a large global problem for solving the rain streaks \mathcal{R} into several smaller, easier-to-solve, local sub-problems of $\mathcal{Y}, \mathcal{S}, \mathcal{X}, \mathcal{T}$ and \mathcal{L} , and then get the the solution of the global problem by coordinating the sub-problems. It can speed up the whole process of the algorithm. And the video derain algorithm flow as shown in the following algorithm.

5 Rain streaks removal experiments

To verify our algorithm’s performance, we selected four methods to compare with ours. There are the method using L_0 gradient minimization technique [1](denoted as 16’L0), the method using layer priors [25](denoted as 16’LP), the method via utilizing discriminatively intrinsic priors [21](denoted as RSR) and the method via joint bi-layer optimization [12](denoted as FastDeRain). At the same time, one of the important characteristics of the rainy video we chose is that the rain in the video is not perpendicular to the ground, but with a certain angle to the ground.

5.1 Real-world Experiments

We select four kinds of rainy video to verify our algorithm better than other four methods. These rainy videos are Pedestrian, Wall, Car and Yard respectively. In addition, the Pedestrian video has the size of $240 \times 360 \times 3 \times 10$, the Yard video has the size of $640 \times 380 \times 3 \times 30$, the Car video and the Wall video have the same size of $360 \times 640 \times 3 \times 10$. In these four kinds of rainy video, we choose one frame from these four videos respectively and randomly to show the derain effect in visual. In addition, because of the lack of the ground truth of the real-world rainy data, we can not directly calculate the index values between the ground truth and the derain images. In this part, we use the visual effects as the only evaluation.

As shown is Fig.3, there are the whole rain removal results of four kinds of real-world data. Firstly, the first, second, third, and fifth rows in the figure are the results of the complete selected frames of the four real-world data respectively. The fourth row corresponds to a partial enlarged view of the

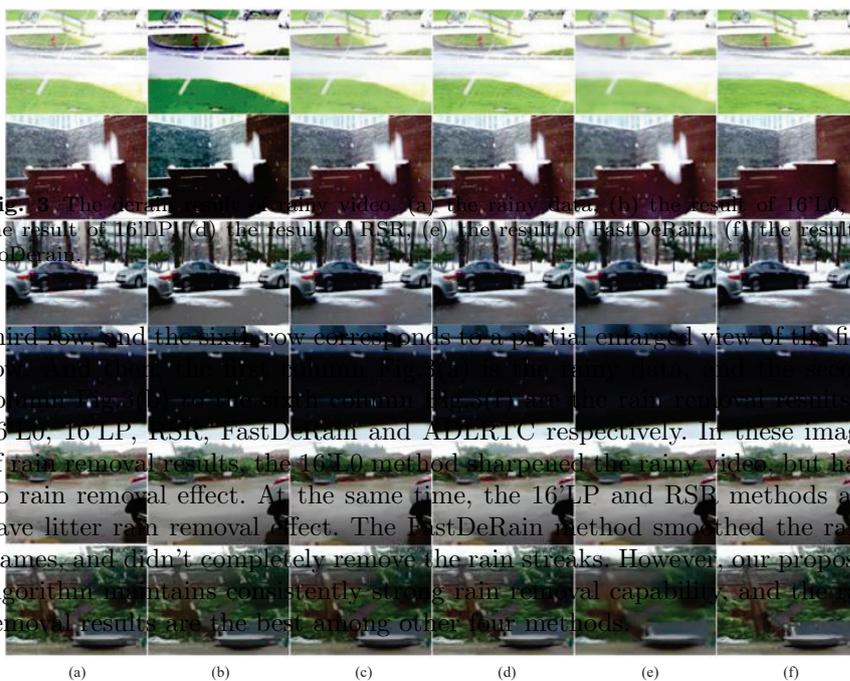


Fig. 3. The rain removal results of rainy videos: (a) the rainy data, (b) the result of 16L0, (c) the result of 16LP, (d) the result of RSR, (e) the result of FastDeRain, (f) the result of RoDeRain.

third row, and the sixth row corresponds to a partial enlarged view of the fifth row. The first row and second row correspond to the original rainy data and rain removal results of 16L0, 16LP, RSR, FastDeRain and ADERIC respectively. In these images of rain removal results, the 16L0 method sharpened the rainy video, but have no rain removal effect. At the same time, the 16LP and RSR methods also have litter rain removal effect. The FastDeRain method smoothed the rainy frames, and didn't completely remove the rain streaks. However, our proposed algorithm maintains consistently strong rain removal capability, and the rain removal results are the best among other four methods.

5.2 Simulated Experiments

In simulated experiments, we also select four kinds of rainy video to verify our algorithm better than other five methods. These rainy videos are City, Mountain, Night and Opera House respectively. In addition, the City video

has the size of $462 \times 762 \times 3 \times 3$, the Mountain video has the size of $462 \times 762 \times 3 \times 3$, the Night video and the Opera House video have the same size of $691 \times 1200 \times 3 \times 3$. In these four kinds of rainy video, we choose 1 frame from these four videos respectively and randomly to show the derain effect in visual. In the simulated experiments, we used the peak signal-to-noise ratio (PSNR) [10], structural similarity index measurement (SSIM) [10], and feature similarity index measurement (FSIM) [16] to verify the effectiveness of the proposed algorithm as a quantitative assessment. We compute the mean value of the three kinds of evaluation indices mentioned above for each frame of rainy videos to compare with different methods, called MPSNR, MSSIM and MFSIM.

In order to better show our rain removal effect, we select a partial enlargement from the complete frames to show our results as shown in Fig.4. So these four rows are the rain results of the four kinds of simulated data. Similarly, the first column Fig.4(a) is the local magnification in selected rainy frame of the simulation data. And the second column Fig.4(b) to the sixth column Fig.4(f) are the rain removal results of 16'L0, 16'LP, RSR, FastDeRain and RoDerain respectively. Unlike real data, there is the ground truth in simulated data as shown is Fig.4(g). With the contrast of the ground truth, we can see our rain removal algorithm's advantages better.

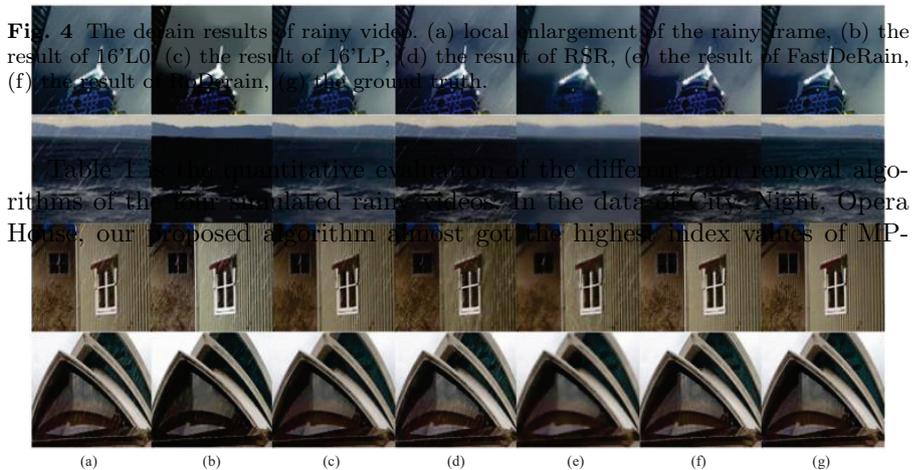


Fig. 4 The derain results of rainy video. (a) local enlargement of the rainy frame, (b) the result of 16'L0, (c) the result of 16'LP, (d) the result of RSR, (e) the result of FastDeRain, (f) the result of RoDerain, (g) the ground truth.

Table 1 shows the quantitative evaluation of the different rain removal algorithms of the four simulated rainy videos. In the data of City, Night, Opera House, our proposed algorithm almost got the highest index values of MP-

Table 1 Quantitative evaluation of the different derain algorithms of four rainy videos

Video	Index	Methods				
		16'L0	16'CVPR	RSR	FastDeRain	Proposed
City	MPSNR	25.2352	20.4635	34.6130	31.9802	35.8856
	MSSIM	0.9301	0.9247	0.9532	0.9869	0.9742
	MFSIM	0.9154	0.9251	0.9594	0.9642	0.9787
Mountain	MPSNR	23.4567	26.1298	30.3163	29.8564	29.7824
	MSSIM	0.7819	0.8705	0.9603	0.9683	0.9745
	MFSIM	0.8463	0.9226	0.9820	0.9849	0.9892
Night	MPSNR	24.0162	16.6175	23.4928	26.0321	30.4106
	MSSIM	0.8421	0.7112	0.8675	0.8806	0.9687
	MFSIM	0.9296	0.8615	0.9329	0.9788	0.9890
Opera_House	MPSNR	26.5367	27.2913	32.1703	29.3660	35.7828
	MSSIM	0.9007	0.9375	0.9666	0.9721	0.9888
	MFSIM	0.9224	0.9533	0.9785	0.9611	0.9938

SNR, MSSIM and MFSIM. At the same time, the visual rain removal results are better than other four methods. Although, in the data of Mountain, the RSR method's the value of MPSNR is slightly higher than ours. However, combining with the visual effects in Fig.4, our proposed algorithm reached the best rain removal results.

6 Conclusion

We propose a novel rotational video derain algorithm via nonconvex and non-smooth algorithm. We introduce the rotation operator, and the constraint of L_1 -norm, L_0 -norm and generalize non-convex non-smooth low-rank minimization to propose a novel algorithm for rain streaks removal by decomposed the rain-free video and the rain streak from the known rainy video. Extensive experiments prove that our model achieved excellent rain removal effects. In particular, our algorithm can remove the rain with a certain inclination angle to the ground. This shows that the rotation operator plays its role well. At the same time, the non-convex and non-smooth adaptive low-rank constraint can improve the quantitative evaluation of the restored video images to a certain extent. In a word, our algorithm is more advanced than existed other rain removal algorithms.

Acknowledgement

This work is supported by the National Natural Science Foundation of China under Grant 61701259, 61572503, 61872424, 6193000388, and NUPTSF (Grant No. NY218001).

References

1. B.N.Manu. Rain removal from still images using l0 gradient minimization technique. *International Conference on Information Technology & Electrical Engineering. IEEE*, 2016.
2. Cai, J.F, Candas, E.J., Shen, and Z. A. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.
3. C.Chang and C.J.Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent System and Technology*, 2(3):1–27, 2012.
4. C.Lu, J.Tang, and S.Yan. Generalized nonconvex nonsmooth low-rank minimization. *Western medicine; the medical journal of the west*, pages 4130–4137, 2014.
5. D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on pattern analysis and machine intelligence*, 25:564–577, 2003.
6. D. L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Inf. Theory*, 41(3):613–627, may 1995.
7. E.A.H.Alaa. A novel approach for rain removal from videos using low-rank recovery. In *Proceedings of 2014 Fifth International Conference on Intelligent Systems, Modelling and Simulation. New York: IEEE Press*, pages 351–356, 2014.
8. E.Ghadimi, A.Teixeira, and I.Shames. Optimal parameter selection for the alternating direction method of multipliers (admm): Quadratic problems. *Automatic Control IEEE Transactions on*, 60(3):644–658, 2015.
9. K. Garg and S. K. Nayar. Vision and rain. *International Journal of Computer Vision*, 75(1):3–27, 2007.
10. Alain HorÁ and D.Ziou. Image quality metrics: Psnr vs. ssim. *2010 International Conference on Pattern Recognition. IEEE Computer Society*, 2010.
11. 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(9):2658–2670, 2015.
12. T. Jiang, T. Huang, X. Zhao, L. Deng, and Y. Wang. Fastderain: A novel video rain streak removal method using directional gradient priors. *IEEE Transactions on Image Processing*, 28(4):2089–2102, 2019.
13. J.Pan and Z.Su. Fast l (0) -regularized kernel estimation for robust motion deblurring. *Western medicine; the medical journal of the west*, 8(2):55, 1967.
14. L.W. Kang, C.W. Lin, and Y.H. Fu. Automatic singleimage-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing*, 21(4):1742–1755, 2012.
15. Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Rain streak removal using layer priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2736–2744, 2016.
16. L.Zhang, L.Zhang, and X.Mou. Fsim: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing*, 20(8):2378, 2011.
17. L.Zhu, C.W.Fu, and D.Lischinski. Joint bi-layer optimization for single-image rain streak removal. *2017 IEEE International Conference on Computer Vision (ICCV). IEEE Computer Society*, 2017.
18. N.Srebro and T.Jaakkola. Weighted low-rank approximations. In *Proceedings of 10th International Conference on Machine Learning. Washington:American Association for Artificial Intelligence*, page 2003, 2003.
19. S.Starik and M.Werman. Simulation of rain in videos. *Texture Workshop. Nice, France:IEEE Press*, 2:406–409, 2003.
20. T.Okatant, T.Yoshida, and K.Deguchi. Efficient algorithm for low-rank matrix factorization with missing components and performance comparison of latest algorithms. In *Proceedings of IEEE International Conference on Computer Vision. New York: IEEE Press*, pages 842–849, 2011.
21. T.X.Jiang, T.Z.Huang, and X.L.Zhao. A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors. *30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR, 2017)*, 2017.
22. X.Cao. Dynamic remote sensing monitoring of land use in nanjing based on tm images. *Geomatics and Information Science of Wuhan University*, 31(11):958–961, 2006.

23. Y.L.Chen and C.T.Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1968–1976, 2013.
24. Y.L.Chen and C.T.Hsu. Proceedings of iee international conference on computer vision. new york: Ieee press. *Geomatics and Information Science of Wuhan University*, pages 1968–1975, 2013.
25. Y.Li, R.T.Tan, and X.Guo. Rain streak removal using layer priors. *Computer Vision and Pattern Recognition. IEEE*, 2016.
26. Weijiang Yu, Zhe Huang, Wayne Zhang, Litong Feng, and Nong Xiao. Gradual network for single image de-raining. In *Proceedings of the 27th ACM International Conference on Multimedia*, page 1795–1804, 2019.
27. Yirui Zeng and Ma Zhengming. A lightweight channel-spatial attention network for real-time image de-raining. In *Proceedings of the 2019 2nd International Conference on Digital Medicine and Image Processing*, page 43–48. Association for Computing Machinery, 2019.
28. Z.Wang, Q.Kang, and Y.Xun. Military reconnaissance application of high-resolution optical satellite remote sensing. In *International Symposium on Optoelectronic Technology & Application: Optical Remote Sensing Technology & Applications*, 2014.
29. Z.Zhu, G.Xu, B.Yang, D.Shi, and X.Lin. Visatram: a real-time vision system for automatic traffic monitoring. *Image and Vision Computing*, 18(10):781–794, 2000.