ACADEMIC
World Academic Union

**ISSN** 1746-7659, England, UK
**Journal of Information and Computing Science**
Vol. 16, No. 1, 2021, *pp*.052-063

# Total variation regularization low-rank decomposition based tensor model for video rain streaks removal

Xinghan Lu[1], Yuhui Zheng, Jianwei Zhang
Nanjing University of Information Science and Technology, Nanjing-210044, China
(Received January 25, 2021, accepted March 17, 2021)

**Abstract:** With superior real-time and storage performance, outdoor computer vision systems have high application value in traffic, public security, identification detection and other fields, but the captured images are affected by environmental factors such as outdoor rainfall, which have obscuration or missing problems and are not conducive to the processing and application of post-level systems. To this end, this paper proposes a tensor model based on total variation regularization low-rank decomposition for video rain streaks removal. Considering the influence of moving objects in the video image on the low-rank structure of the video background, the rainy video is decomposed into static background, dynamic objects and rain streaks, and their a priori characteristics are analyzed separately, combined with the corresponding low-rank characteristics or sparse characteristics to construct a tensor model, and the targets are extracted through low-rank decomposition, and then the rain removal is completed. The proposed tensor model is solved by the alternating direction multiplier method (ADMM), and extensive experiments are carried out on synthetic and real data sets. The results show that the proposed method can effectively remove rain streaks from video images while retaining more background details under dynamic background conditions. Compared with related advanced methods, the proposed method has advantages in three comprehensive quantifiers, namely, peak signal-to-noise ratio, structural similarity and residuals.

**Keywords:** Rain removal, tensor model, total variation, low rank, ADMM.

## 1. Introduction

Outdoor computer vision systems have a wide range of applications in many fields such as road traffic and public safety video surveillance. However, rain can degrade the acquired video images, resulting in image contrast degradation, blurring or detail loss, etc., which adversely affects the subsequent work of computer vision systems (e.g., target detection [1,2], recognition [3], and tracking [4]). Therefore, it is of great research significance to recover such video images to minimize the effect of rain on images and to improve the stability and practicality of outdoor computer vision systems.

In recent years, many methods have been used to de-rain video images. From the perspective of methods, there are three main categories: time-domain-based methods, frequency-domain-based methods and sparse-domain-based methods.

Based on the time-domain perspective, Zhu et al. [5] used a photometric model to obtain the candidate rain streaks and used the inter-frame difference method for motion object detection, deducting the motion objects in the candidate rain streaks to obtain the final rain line detection results. The inter-frame difference method has advantages such as simple computation, but it is susceptible to background illumination changes and extraneous events, resulting in large errors in motion detection. For this reason, Wang et al. [6] used a multi-frame anisotropic filter based on a kernel function that can adaptively change the filter intensity and direction according to the local characteristics of the rain streaks to remove the rain streaks while the rain streaks are detected and improve the robustness in dynamic scenes.

Based on the frequency domain perspective, Barnum et al. [7], abandoned the approach of analyzing pixels and pixel blocks from the time domain and proposed a video de-rain method based on frequency domain spatial analysis. Santhaseelan et al. [8] proposed a video de-rain method based on local phase consistency, which overcomes the error brought by the fixed threshold recovery model and enhances the smoothness of the video. Chen et al. [9] obtained detail edge information based on wavelet transform, filtered after doing difference operation using inter-frame pixel brightness, and used fast bilateral filtering for rain line removal.

Based on the sparse domain perspective, Kang et al. [10] used bilateral filtering to decompose the image with rain into low-frequency and high-frequency components based on morphological component

analysis (MCA), and then separated the rain streaks in the high-frequency components by dictionary learning and sparse coding. The sparse decomposition-based rain removal method does not require contextual information and has the advantages of wide applicability, from which a series of improved methods have been derived [11-17]. To improve the convergence speed of the algorithm, Ramya et al. [13] used an enhanced K-SVD (EKSVD) method for dictionary learning and an orthogonal matching tracking method for sparse coding of images with rain. Li et al. [18] used sub-block image a priori information for modeling, and used Gaussian Mixture Model (GMM) for learning the rain line component and the video background, the background part is further constrained. This method can remove rain streaks better. However, when the input image contains many structures similar to the distribution of rain streaks, this method is difficult to effectively distinguish between layers with and without rain. KIM et al. [20] proposed a time-dependent video de-rain method based on obtaining the initial rain map by optical flow estimation, applying a support vector machine classifier refinement, and eliminating rain streaks using a low-rank matrix filling technique [21]. Li et al. [23] learned a multiscale convolution filter from rain data to decompose the rain component into different levels of rain streaks and sparse coding of the different features. The method achieves excellent results in real videos with rain, but the rain removal results may be blurred if there are complex moving objects in the video.

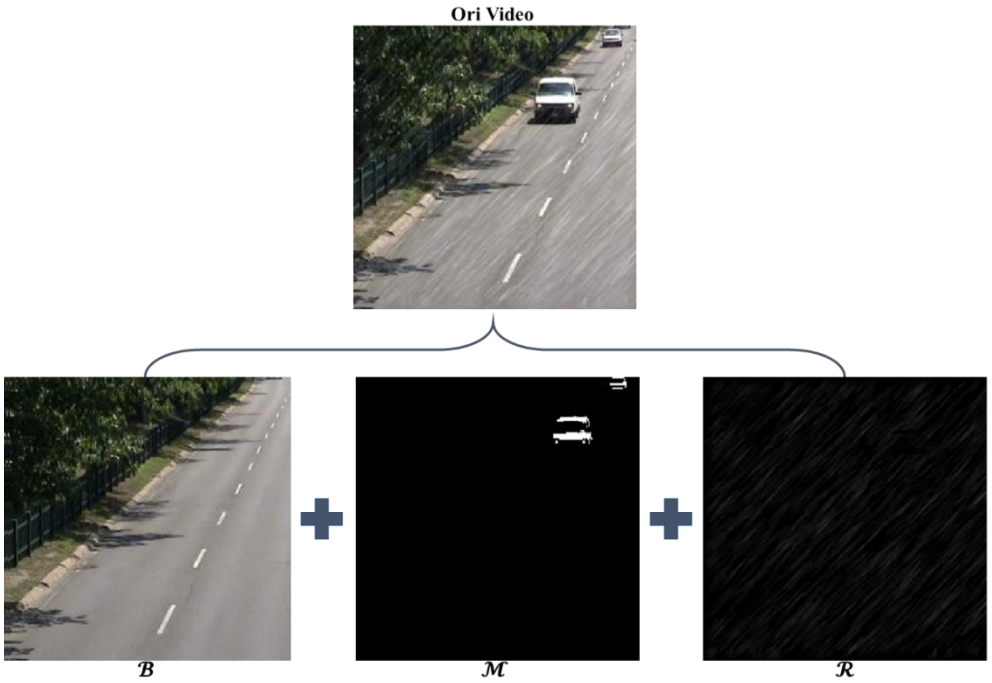


Fig.1 Rainy video $\mathcal{O}$, static background $\mathcal{B}$, moving object $\mathcal{M}$ and rain streaks $\mathcal{R}$

Unlike the methods that require dictionary learning, Chen et al. [19] proposed a tensor low-rank epistemic model to detect spatio-temporal correlated rain streaks using similar patterns of rain streaks in the image scene. The method does not require dictionary learning, but it causes certain rain line misclassification due to the insufficiently strong constraints. Jiang et al. [22] proposed a tensor-based video rain line removal model with great advantages in time consumption by considering the difference between the intrinsic properties of video background and rain marks and enhancing sparsity using the parametric norm. Sun et al. [24] proposed a directional norm-based de- rain tensor model, which simulates the non-vertical landing of real rain streaks with constraints on the rain line direction. The method can effectively remove the rain streaks at different angles, but it is not optimistic in terms of running time. Considering the different prior information in the rainy video, Wang et al. [25] proposed a group sparsity-based rain removal method, which enhances the sparsity of rain streaks using group sparsity while modeling different prior information to facilitate rain line separation.

Recently, deep learning-based methods have also achieved better results [26-30]. It is worth noting that the rain removal effect depends heavily on the number and diversity of training datasets. To make the rain removal method more flexible and stable, inspired by the literature [25], a video rain removal tensor model based on fully variational regular low-rank decomposition is proposed in this paper, considering the irregular motion of dynamic objects in the video background.

The remaining sections of this paper are organized as follows: the first section analyzes and processes the a priori information of the rainy video. Section II gives the proposed model and algorithm. Section III shows the experimental results and the comparison of quantitative metrics. Finally, Section IV concludes the paper with an outlook.

## 2. Problem Formulation

Considering the moving objects in the video, the K-frame rainy video $\mathcal{O} = \{f_k\}_{k=1}^K$ is further decomposed into a linear overlay as follows：

$$\mathcal{O} = \mathcal{B} + \mathcal{M} + \mathcal{R}, \tag{1}$$

where $\mathcal{B} = \{b_k\}_{k=1}^K$, $\mathcal{M} = \{m_k\}_{k=1}^K$ and $\mathcal{R} = \{r_k\}_{k=1}^K$ are static background, moving object and rain line respectively, see Fig.1.

The goal of rain removal is to decompose the rain-free video $\mathcal{B} + \mathcal{M}$ without the rain line $\mathcal{R}$ from the input rain-bearing video $\mathcal{O}$. To solve this discomfort problem, we first need to analyze their a priori information and construct a model by combining the corresponding low rankness and sparsity.

**The priori analysis of rain streaks $\mathcal{R}$:** Normally, rain streaks are often considered to be sparse. To enhance the sparsity of rain streaks and facilitate their separation, the $L_1$ norm regular term $\|\mathcal{R}\|_1$ is used to model the sparsity of rain streaks in the spatio-temporal domain. In general, the rain streaks fall in similar directions and are approximately vertical, which has a limited effect on the vertical gradient. As shown in Fig.2, it can be seen that the vertical gradient of the rain line is much sparser compared to the clean video. Therefore, the $L_1$ parametric number $\|\nabla_x \mathcal{R}\|_1$ is used to enhance the sparsity of the rain streaks.

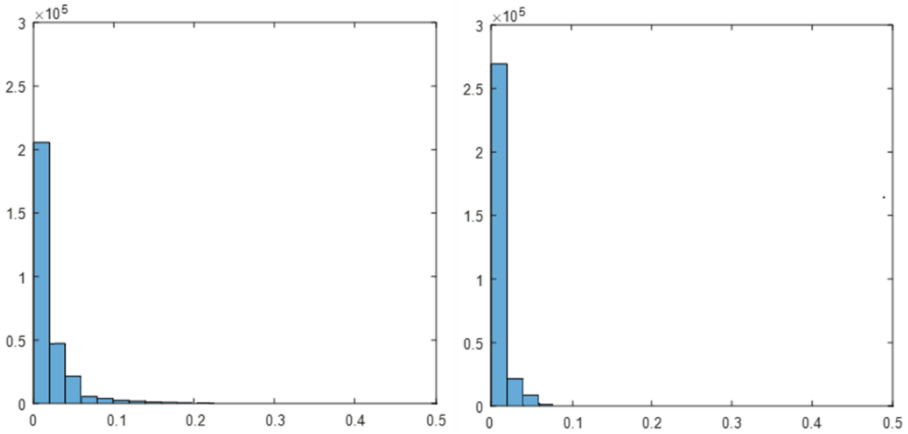


Fig.2 Histogram of vertical gradient of clean video $\mathcal{B}$ and rain streaks $\mathcal{R}$

**The priori analysis of static background $\mathcal{B}$:** Converting each frame of the video into a column vector and arranging them in a matrix, we have:

$$\mathcal{O} = [vec(f_1)|\cdots|vec(f_K)],$$
$$\mathcal{B} = [vec(b_1)|\cdots|vec(b_K)],$$
$$\mathcal{M} = [vec(m_1)|\cdots|vec(m_K)],$$
$$\mathcal{R} = [vec(r_1)|\cdots|vec(r_K)].$$

For each frame in the video, the static background $\mathcal{B}$ has a similar structure with low-rankness, so the constructed matrix $\mathcal{B}$ is a low-rank matrix. And for the dynamic object matrix $\mathcal{M}$, after rearranging the same motion components in the frames by some appropriate alignment transformations, low-rankness can also be obtained and a low-rank matrix is constructed.

**The priori analysis of moving object $\mathcal{M}$:** There are often various irregular dynamic objects in a video, and their motion can cause changes in the intensity of the video background, resulting in misclassification of static background components with rain streaks and affecting the removal of rain streaks. Total variation regularization has been widely used in image and video processing for its superior performance in suppressing discontinuous variations [31-33]. It is a good choice to employ it to suppress intensity variations caused by dynamic backgrounds and to fill the gaps caused by dynamic objects [34]. Let $\mathcal{B}(x, y, t)$ denote the intensity for $(x, y)$ at time $t$. Use

$$\mathcal{M}_h(x, y, t) = \mathcal{M}(x + 1, y, t) - \mathcal{M}(x, y, t),$$
$$\mathcal{M}_v(x, y, t) = \mathcal{M}(x, y + 1, t) - \mathcal{M}(x, y, t),$$

$$\mathcal{M}_t(x,y,t) = \mathcal{M}(x,y,t+1) - \mathcal{M}(x,y,t),$$

to denote the results of the three difference operations with periodic boundary conditions in the horizontal, vertical and temporal directions for $(x,y)$ at time $t$, respectively. Let $D_h m = \mathrm{vec}(\mathcal{M}_h)$, $D_v m = \mathrm{vec}(\mathcal{M}_v)$ and $D_t m = \mathrm{vec}(\mathcal{M}_t)$ denote the vectorization of the three difference operations, respectively. Set $Dm = [D_h m^T, D_v m^T, D_t m^T]^T$ denotes series differential operation, where $D \in R^{3d_v d_h d_t \times d_v d_h d_t}$ The $i$-th element in $D_h m$, $D_v m$ and $D_t m$, i.e. $[D_h m]_i$, $[D_v m]_i$ and $[D_t m]_i$, describes the intensity change at the $i$-th point in $m$ along the horizontal, vertical, and temporal directions, and the change in intensity can be quantified using the $L_1$ norm of $[[D_h m]_i, [D_v m]_i, [D_t m]_i]^T$. The anisotropic total variational norm is obtained by summing over all vector norms at different points as follows,

$$\|\mathcal{M}\|_{TV} = \sum_i (|[D_h m]_i| + |[D_v m]_i| + |[D_t m]_i|). \tag{2}$$

Facilitate the separation of dynamic objects in the video background by minimizing $\|\mathcal{M}\|_{TV}$.

# 3. The proposed method and algorithm

As a summary of the discussion of prior and regular terms, the model in this paper can be succinctly formulated in the following form:

$$\begin{aligned} &\min_{\mathcal{B},\mathcal{M},\mathcal{R}} rank(\mathcal{B}) + \alpha_1 rank\big(P(\mathcal{M})\big) + \alpha_2\|\mathcal{M}\|_{TV} + \alpha_3\|\nabla_x\mathcal{R}\|_1 + \alpha_4\|\mathcal{R}\|_1, \\ &s.t.\, \mathcal{O} = \mathcal{B} + \mathcal{M} + \mathcal{R}, \mathcal{O} \geq \mathcal{B} \geq 0, \mathcal{O} \geq \mathcal{M} \geq 0, \mathcal{O} \geq \mathcal{R} \geq 0, \end{aligned} \tag{3}$$

where the operator $P(\cdot)$ represents some transformations (e.g., translation and rotation) performed on the local components of the video, which provide the basis for extracting the latent similar structure of the moving objects.

For model (3) it is difficult to solve directly, so the solution process can be divided into two steps,

(1) Extract the low-rank component $\mathcal{B}$ from $\mathcal{O}$:

$$\begin{aligned} &\min_{\mathcal{B},L}\|\mathcal{B}\|_* + \alpha_1\|\mathcal{L}\|_1 + \alpha_2\|\mathcal{L}\|_{TV} \\ &\min_{\mathcal{B},L}\|\mathcal{B}\|_* + \alpha_1\|\mathcal{L}\|_1 + \alpha_2\|\mathcal{L}\|_{TV}, \\ &s.t.\ \mathcal{O} = \mathcal{B} + \mathcal{L}, \mathcal{O} \geq \mathcal{B} \geq 0, \mathcal{O} \geq \mathcal{L} \geq 0; \end{aligned}$$

(2) Extract the aligned transformed low-rank component $\mathcal{M}$ from $\mathcal{L}$:

$$\begin{aligned} &\min_{\mathcal{M},\mathcal{R}}\|P(\mathcal{M})\|_* + \alpha_3\|\nabla_x\mathcal{R}\|_1 + \alpha_4\|\mathcal{R}\|_1, \\ &s.t.\, \mathcal{L} = \mathcal{M} + \mathcal{R}, \mathcal{L} \geq M \geq 0, \mathcal{L} \geq \mathcal{R} \geq 0. \end{aligned}$$

## 3.1 Extraction of static background $\mathcal{B}$

The subproblem for static background $\mathcal{B}$ is as follows:

$$\begin{aligned} &\min_{\mathcal{B},L}\|\mathcal{B}\|_* + \alpha_1\|\mathcal{L}\|_1 + \alpha_2\|\mathcal{L}\|_{TV}, \\ &s.t.\ \mathcal{O} = \mathcal{B} + \mathcal{L}, \mathcal{O} \geq \mathcal{B} \geq 0, \mathcal{O} \geq \mathcal{L} \geq 0, \end{aligned} \tag{4}$$

here $\mathcal{L}$ contains the moving foreground and rain streak components, i.e. $\mathcal{L} = \mathcal{M} + \mathcal{R}$. In the ADMM framework, equation (4) can be reformulated into the following equivalence constraint form:

$$\begin{aligned} &\min_{\mathcal{X},\mathcal{N},\mathcal{V}}\|\mathcal{X}\|_* + \alpha_1\|\mathcal{N}\|_1 + \alpha_2\|\mathcal{V}\|_{TV}, \\ &\text{s.t.}\, \mathcal{X} = \mathcal{B}, \mathcal{N} = \mathcal{O} - \mathcal{B}, \mathcal{V} = \mathcal{O} - \mathcal{B}, \mathcal{O} \geq \mathcal{B} \geq 0, \end{aligned} \tag{5}$$

The augmented Lagrangian function $\mathcal{L}_{\mathcal{A}}$ is of the following form:

$$\begin{aligned} \mathcal{L}_{\mathcal{A}} = \|\mathcal{X}\|_* + \alpha_1\|\mathcal{N}\|_1 + \alpha_2\|\mathcal{V}\|_{TV} + \langle\Lambda_1, \mathcal{B} - \mathcal{X}\rangle + \frac{\beta_1}{2}\|\mathcal{B} - \mathcal{X}\|_F^2 + \langle\Lambda_2, \mathcal{O} - \mathcal{B} - \mathcal{N}\rangle \\ + \frac{\beta_2}{2}\|\mathcal{O} - \mathcal{B} - \mathcal{N}\|_F^2 + \langle\Lambda_3, \mathcal{O} - \mathcal{B} - \mathcal{V}\rangle + \frac{\beta_3}{2}\|\mathcal{O} - \mathcal{B} - \mathcal{V}\|_F^2, \end{aligned} \tag{6}$$

where $\Lambda = [\Lambda_1, \Lambda_2, \Lambda_3]$ are Lagrange multipliers, and $\beta = [\beta_1, \beta_2, \beta_3]$ are positive parameters. Therefore, this joint minimization problem can be decomposed into four subproblems, i.e. $\mathcal{X}$, $\mathcal{N}$, $\mathcal{V}$ and $\mathcal{B}$, and solved separately using the ADMM algorithm.

**$\mathcal{X}$-subproblem:** Subproblem $\mathcal{X}$ can be formulated as follows:

$$\min_{\mathcal{X}}\|\mathcal{X}\|_* + \frac{\beta_1}{2}\left\|\mathcal{B} - \mathcal{X} + \frac{\Lambda_1}{\beta_1}\right\|_F^2, \tag{7}$$

The tensor kernel parametrization is defined as

$$\|\mathcal{X}\|_* = \sum_{i=1}^{n}\|\boldsymbol{X}_i\|_*, \boldsymbol{X}_i = \mathrm{Unfold}_i(\mathcal{S}).$$

Thus the solution of equation (7) can be expressed as

$$\mathcal{X}^{(t+1)} = \sum_{i=1}^{3} \frac{1}{3} \mathrm{Fold}_i\left(\boldsymbol{X}_{(i)}^{(t+1)}\right), \tag{8}$$

here $\boldsymbol{X}_{(i)}^{(t+1)} = \mathcal{Q}_{\frac{1}{\beta_1}}\left(\mathcal{B}_{(i)}^{(t)} + \Lambda_{1(i)}^{(t)}/\beta_1\right)$ $(i = 1,2,3)$, which $\mathcal{Q}_{1/\beta_1}$ denotes soft thresholding of singular values of $\mathcal{X}$.

**$\mathcal{N}$-subproblem:** With other parameters fixed, the subproblem $\mathcal{N}$ can be transformed into

$$\min_{\mathcal{N}} \lambda\|\mathcal{N}\|_1 + \frac{\beta_2}{2}\left\|(\mathcal{O} - \mathcal{B}) - \mathcal{N} + \frac{\Lambda_2}{\beta_2}\right\|_F^2, \tag{9}$$

The closed solution can be obtained through the soft threshold operator problem (9) as follows:

$$\mathcal{N}^{(t+1)} = \mathrm{Soft}_{\frac{\alpha_1}{\beta_2}}\left(\left(\mathcal{O} - \mathcal{B}^{(t)}\right) + \frac{{\Lambda_2}^{(t)}}{\beta_2}\right). \tag{10}$$

**$\mathcal{V}$-subproblem:** Subproblem $\mathcal{V}$ can be formulated as follows:

$$\begin{aligned}\mathcal{V}^{(t+1)} &= \mathrm{argmin}\,\alpha_2\left\|\mathcal{V}^{(t)}\right\|_{TV} + \frac{\beta_3}{2}\left\|\left(\mathcal{O} - \mathcal{B}^{(t)}\right) - \mathcal{V} + \frac{{\Lambda_3}^{(t)}}{\beta_3}\right\|_F^2 \\ &= \mathrm{argmin}\,\alpha_2\|Dv\|_q + \frac{\beta_3}{2}\left\|\left(\mathcal{O} - \mathcal{B}^{(t)}\right) - \mathcal{V} + \frac{{\Lambda_3}^{(t)}}{\beta_3}\right\|_F^2.\end{aligned} \tag{11}$$

To solve the problem (11), the auxiliary variable $\mathcal{K} \in \mathbb{R}^{3d_h d_v d_t \times 1}$ is introduced to replace $Dv$, we have

$$\mathcal{V}^{(t+1)} = \mathrm{argmin}\,\alpha_2\|\mathcal{K}\|_q + \frac{\beta_3}{2}\left\|\left(\mathcal{O} - \mathcal{B}^{(t)}\right) - \mathcal{V} + \frac{{\Lambda_3}^{(t)}}{\beta_3}\right\|_F^2, \mathrm{s.t.}\,\mathcal{K} = Dv. \tag{12}$$

The augmented Lagrangian function is as follows:

$$\mathcal{L}_{\mathcal{A}}(\mathcal{V}, \mathcal{K}, \mu) = \alpha_2\|\mathcal{K}\|_q + \frac{\beta_3}{2}\left\|\mathcal{O} - \mathcal{B} - \mathcal{V} + \frac{\Lambda_3}{\beta_3}\right\|_F^2 + \frac{\lambda}{2}\left\|\mathcal{K} - Dv - \frac{\mu}{\lambda}\right\|_F^2, \tag{13}$$

where $\lambda$ is the same as $\beta_3$ and is a positive parameter and $\mu$ is a Lagrangian multiplier. Fix other variables, update $\mathcal{V}^{(t+1)}$, we have

$$\mathcal{V}^{(t+1)} = \mathrm{argmin}\,\mathcal{L}_{\mathcal{A}}\left(\mathcal{V}, \mathcal{K}^{(t)}, \mu^{(t)}\right). \tag{14}$$

Considering its normal equation, we have

$$(\beta_3 I + \lambda D^T D)v = \mathcal{Q}, \mathcal{V}^{(t+1)} = \mathrm{reshape}(v), \tag{15}$$

here,

$$\mathcal{Q} = \beta_3\,\mathrm{vec}\left(\mathcal{O} - \mathcal{B}^{(t)} + \frac{\Lambda_3}{\beta_3}\right) + \lambda\left(D^T\mathcal{K}^{(t)} + D^T\frac{\mu^{(t)}}{\lambda}\right). \tag{16}$$

Due to the block-circulant structure of the matrix, it can be diagonalized by a 3-D-DFT matrix [35]. Thus, $v^{(t+1)}$ can be obtained from the following equation:

$$\mathcal{F}^{-1}\left(\frac{\mathcal{F}(\mathcal{Q})}{\beta_3 1 + \lambda(|\mathcal{F}(D_h)|^2 + |\mathcal{F}(D_v)|^2 + |\mathcal{F}(D_t)|^2)}\right), \tag{17}$$

where $\mathcal{F}(\cdot)$ 3D Fourier transform. Fix other variables, update $\mathcal{K}^{(t+1)}$, we have

$$\mathcal{K}^{(t+1)} = \mathrm{argmin}\,\mathcal{L}_2\left(\mathcal{V}^{(t+1)}, \mathcal{K}, \mu^{(t)}\right). \tag{18}$$

The iterative formula of K is as follows:

$$\mathcal{K}^{(t+1)} = \mathrm{Soft}_{\frac{\alpha_2}{\lambda}}\left(Dv^{(t+1)} + \frac{\mu^{(t)}}{\lambda}\right), \tag{19}$$

the multiplier $\mu$ is updated via:

$$\mu^{(t+1)} = \mu^{(t)} + \lambda\left(\mathcal{K}^{(t+1)} - Dv^{(t+1)}\right). \tag{20}$$

**$\mathcal{B}$-subproblem:** Finally, subproblem $\mathcal{B}$ has the following form:

$$\min_{\mathcal{B}} \frac{\beta_1}{2}\left\|\mathcal{X} - \mathcal{B} - \frac{\Lambda_1}{\beta_1}\right\|_F^2 + \frac{\beta_2}{2}\left\|\mathcal{N} - (\mathcal{O} - \mathcal{B}) - \frac{\Lambda_2}{\beta_2}\right\|_F^2 + \frac{\beta_3}{2}\left\|\mathcal{V} - (\mathcal{O} - \mathcal{B}) - \frac{\Lambda_3}{\beta_2}\right\|_F^2. \tag{21}$$

It exists as a closed solution,

$$\mathcal{B}^{(t+1)} = \mathcal{F}^{-1}\left(\mathcal{F}(K_1)/\mathcal{F}(K_2)\right), \tag{22}$$

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

$$\begin{aligned}K_1 &= \beta_1\mathcal{X}^{(t+1)} - {\Lambda_1}^{(t)} + \beta_2\left(\mathcal{O} - \mathcal{N}^{(t+1)}\right) + {\Lambda_2}^{(t)} + \beta_3\left(\mathcal{O} - \mathcal{V}^{(t+1)}\right) + {\Lambda_3}^{(t)}, \\ K_2 &= \mathcal{I} + \beta_1\mathcal{I} + \beta_2\mathcal{I}.\end{aligned} \tag{23}$$

In the iterative process, the iterative formula for the Lagrange multipliers $\Lambda$ are updated using

$${\Lambda_1}^{(t+1)} = {\Lambda_1}^{(t)} + \beta_1\left(\mathcal{B}^{(t+1)} - \mathcal{X}^{(t+1)}\right),$$

$$\Lambda_2{}^{(t+1)} = \Lambda_2{}^{(t)} + \beta_2\left(\mathcal{O} - \mathcal{B}^{(t+1)} - \mathcal{N}^{(t+1)}\right),$$
$$\Lambda_3{}^{(t+1)} = \Lambda_3{}^{(t)} + \beta_3\left(\mathcal{O} - \mathcal{B}^{(t+1)} - \mathcal{V}^{(t+1)}\right). \quad (24)$$

### 3.2 Extraction of static background $\mathcal{M}$

After extracting the static background $\mathcal{B}$, the obtained $\mathcal{L} = \mathcal{M} + \mathcal{R}$ component is a linear superposition of the moving object and the rain line. According to equation (3), the subproblem of $\mathcal{M}$ extraction can be expressed as

$$\min_{\mathcal{M},\mathcal{R}} \|P(\mathcal{M})\|_* + \alpha_3\|\nabla_x\mathcal{R}\|_1 + \alpha_4\|\mathcal{R}\|_1,$$
$$s.t.\mathcal{L} = \mathcal{M} + \mathcal{R}, \mathcal{L} \geq M \geq 0, \mathcal{L} \geq \mathcal{R} \geq 0. \quad (25)$$

For M in equation (25), an alignment transformation is required to obtain the low-rank structure. Since the motion of the objects is different between frames, global estimation of $P(\cdot)$ is not realistic, so the block matching method is used to solve this problem.

The matrix form L is reshaped into the frame sequence form, and the method in DECOLOR [36] is applied to segment these dynamic components to obtain the moving object $\mathcal{M} = \{m_k\}_{k=1}^{K}$. For each frame $m_k$, blocks are generated as the minimum boundary of the moving object, and pixels similar to the sample block of the current frame are found in each frame. A large number of block matching algorithms for motion estimation exist [37-40], and here a fast and efficient sequential similarity detection algorithm [40] (SSDA) is used, for a block of $n_x \times n_y$ size, we have

$$\underset{x_0,y_0}{\mathrm{SSDA}}(x,y) = \left|p_{k+1}(x_0 + x + i, y_0 + y + j) - \overline{S_{\iota,J}} - p_k(x_0 + i, y_0 + j) + \bar{T}\right|, \quad (26)$$
$$\overline{S_{\iota,J}} = \frac{1}{n_x \times n_y}\sum_{i=0}^{n_x}\sum_{J=0}^{n_y} p_{k+1}(x_0 + x + i, y_0 + y + j),$$
$$\bar{T} = \frac{1}{n_x \times n_y}\sum_{i=0}^{n_x}\sum_{J=0}^{n_y} p_k(x_0 + i, y_0 + j),$$

where $(x_0, y_0)$ and $(x_0 + x, y_0 + y)$ indicate the module block and the block to be compared. Assuming that there are $m$ blocks that are similar in the time domain, the pixels of each matching block are formed into each column of a matrix $P_{j,k}$ of size $(n_x \times n_y) \times m$. Thus, $P$ can be expressed as

$$P_{j,k} = \left(P_{1,j,k}, P_{2,j,k}, \ldots, P_{m,j,k}\right), P_{i,j,k} \in R^2, \quad (27)$$

then,

$$P_{j,k} = M_{j,k} + R_{j,k}. \quad (28)$$

It follows that the column vectors in $M_{j,k}$ have a similar structure and the matrix $\mathcal{M}$ can be viewed as a low-rank matrix. In this way the problem of extracting the motion object $\mathcal{M}$ can be equated to

$$\min_{\mathcal{M},\mathcal{R}} \|\mathcal{M}\|_* + \alpha_3\|\nabla_x\mathcal{R}\|_1 + \alpha_4\|\mathcal{R}\|_1,$$
$$s.t.\mathcal{P} = \mathcal{M} + \mathcal{R}, \mathcal{P} \geq M \geq 0, \mathcal{P} \geq \mathcal{R} \geq 0. \quad (29)$$

In the ADMM framework, equation (29) is reformulated into the following equivalence constraint form:

$$\min_{\mathcal{S},\mathcal{Y},\mathcal{T}} \|\mathcal{S}\|_* + \alpha_3\|\mathcal{Y}\|_1 + \alpha_4\|\mathcal{T}\|_1,$$
$$s.t.\mathcal{S} = \mathcal{M}, \mathcal{Y} = \nabla_x(\mathcal{P} - \mathcal{M}), \mathcal{T} = \mathcal{P} - \mathcal{M}, \mathcal{P} \geq \mathcal{M} \geq 0. \quad (30)$$

The augmented Lagrangian $\mathcal{L}_{\mathcal{A}}$ function can be expressed as:

$$\mathcal{L}_{\mathcal{A}} = \|\mathcal{S}\|_* + \alpha_3\|\mathcal{Y}\|_1 + \alpha_4\|\mathcal{T}\|_1 + \langle\Lambda_4, \mathcal{M} - \mathcal{S}\rangle + \frac{\beta_4}{2}\|\mathcal{M} - \mathcal{S}\|_F^2$$
$$+\langle\Lambda_5, \nabla_x(\mathcal{P} - \mathcal{M}) - \mathcal{Y}\rangle + \frac{\beta_5}{2}\|\nabla_x(\mathcal{P} - \mathcal{M}) - \mathcal{Y}\|_F^2$$
$$+\langle\Lambda_6, (\mathcal{P} - \mathcal{M}) - \mathcal{T}\rangle + \frac{\beta_6}{2}\|(\mathcal{P} - \mathcal{M}) - \mathcal{T}\|_F^2. \quad (31)$$

Therefore, this joint minimization problem can be decomposed into four subproblems, i.e. $\mathcal{S}, \mathcal{Y}, \mathcal{T}$ and $\mathcal{M}$, which solved separately using the ADMM algorithm.

***$\mathcal{S}$*-subproblem:** The subproblem $\mathcal{S}$ can be formulated as:

$$\min_{\mathcal{S}} \|\mathcal{S}\|_* + \frac{\beta_4}{2}\left\|\mathcal{M} - \mathcal{S} + \frac{\Lambda_4}{\beta_4}\right\|_F^2. \quad (32)$$

The tensor kernel norm is defined as $\|\mathcal{S}\|_* = \sum_{i=1}^{n}\|\boldsymbol{S}_i\|_*$, $\boldsymbol{S}_i = \mathrm{Unfold}_i(\mathcal{S})$. The solution of the above equation can be expressed as

$$\mathcal{S}^{(t+1)} = \sum\nolimits_{i=1}^{3}\frac{1}{3}\mathrm{Fold}_i\left(\boldsymbol{S}_{(i)}^{(t+1)}\right), \quad (33)$$

where $\boldsymbol{S}_{(i)}^{(t+1)} = \mathcal{Q}_{\frac{1}{\beta_4}}\left(\mathcal{M}_{(i)}^{(t)} + {\Lambda_4}_{(i)}^{(t)}/\beta_4\right), (i = 1,2,3)$, $\mathcal{Q}_{1/\beta_4}$ denotes soft thresholding of the singular values of $\boldsymbol{S}$.

**$\mathcal{Y}$, $\mathcal{T}$-subproblems:** With other parameters fixed, the subproblems $\mathcal{Y}$ and $\mathcal{T}$ can be transformed as:

$$\min_{\mathcal{Y}} \alpha_3\|\mathcal{Y}\|_1 + \frac{\beta_5}{2}\left\|\nabla_x(\mathcal{P}-\mathcal{M}) - \mathcal{Y} + \frac{\Lambda_5}{\beta_5}\right\|_F^2, \tag{34}$$

$$\min_{\mathcal{T}} \alpha_4\|\mathcal{T}\|_1 + \frac{\beta_6}{2}\left\|(\mathcal{P}-\mathcal{M}) - \mathcal{T} + \frac{\Lambda_6}{\beta_6}\right\|_F^2. \tag{35}$$

The problem (34) and (35) can be obtained as a closed solution by the following soft threshold operator:

$$\mathcal{Y}^{(t+1)} = \mathrm{Soft}_{\frac{\alpha_3}{\beta_5}}\left(\nabla_x\left(\mathcal{P}-\mathcal{M}^{(t)}\right) + \frac{{\Lambda_5}^{(t)}}{\beta_5}\right), \tag{36}$$

$$\mathcal{T}^{(t+1)} = \mathrm{Soft}_{\frac{\alpha_4}{\beta_6}}\left(\left(\mathcal{P}-\mathcal{M}^{(t)}\right) + \frac{{\Lambda_6}^{(t)}}{\beta_6}\right). \tag{37}$$

**$\mathcal{M}$-subproblems:** Finally, the subproblem $\mathcal{M}$ has the following form:

$$\min_{\mathcal{M}} \frac{\beta_4}{2}\left\|\mathcal{M} - \mathcal{S} + \frac{\Lambda_4}{\beta_4}\right\|_F^2 + \frac{\beta_5}{2}\left\|\nabla_x(\mathcal{P}-\mathcal{M}) - \mathcal{Y} + \frac{\Lambda_5}{\beta_5}\right\|_F^2 + \frac{\beta_6}{2}\left\|(\mathcal{P}-\mathcal{M}) - \mathcal{T} + \frac{\Lambda_6}{\beta_6}\right\|_F^2, \tag{38}$$

it has a closed solution

$$\mathcal{M}^{(t+1)} = \mathcal{F}^{-1}\left(\mathcal{F}(K_1)/\mathcal{F}(K_2)\right), \tag{39}$$

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

$$\begin{aligned} K_1 &= \beta_4\mathcal{S}^{(t+1)} - {\Lambda_4}^{(t)} + \nabla_x^T\left(\beta_5\nabla_x\mathcal{P} - \beta_5\mathcal{Y}^{(t+1)} + {\Lambda_5}^{(t)}\right) + \beta_6\left(\mathcal{O} - \mathcal{T}^{(t+1)}\right) + {\Lambda_6}^{(t)}, \\ K_2 &= (1+\beta_6)\mathcal{I} + \beta_5\nabla_x^T\nabla_x. \end{aligned} \tag{40}$$

In the iterative process, the iterative formula for the Lagrange multipliers $\Lambda$ are updated using

$$\begin{aligned} {\Lambda_4}^{(t+1)} &= {\Lambda_4}^{(t)} + \beta_4\left(\mathcal{M}^{(t+1)} - \mathcal{S}^{(t+1)}\right), \\ {\Lambda_5}^{(t+1)} &= {\Lambda_5}^{(t)} + \beta_5\left(\nabla_x\left(\mathcal{P}-\mathcal{M}^{(t+1)}\right) - \mathcal{Y}^{(t+1)}\right), \\ {\Lambda_6}^{(t+1)} &= {\Lambda_6}^{(t)} + \beta_6\left(\mathcal{P}-\mathcal{M}^{(t+1)} - \mathcal{T}^{(t+1)}\right). \end{aligned} \tag{41}$$

**Algorithm 2 TVLSDerain**

**Input：The rainy video** $\mathcal{O}$;
1：**Initialization：** $\mathcal{B}^0 = \mathcal{X}^0 = \mathcal{L}^0 = \mathcal{V}^0 = \mathcal{N}^0 = \mathcal{M}^0 = \mathcal{S}^0 = \mathcal{Y}^0 = \mathcal{T}^0 = 0$;
2：**while** not converged **do**
3： Update $\mathcal{X}$ via Eq. (8); Update $\mathcal{N}$ via Eq. (10);
4： while not converged do
5： Initialization: $\mathcal{K}^0 = 0$;
6： Update $\mathcal{V}$ via Eq. (17); Update $\mathcal{K}$ via Eq. (18); Update multipliers via Eq. (19);
7： end
8： Update $\mathcal{B}$ via Eq. (22); Update multipliers via Eq. (24);
9：**end**
10：Get aligned blocks $\mathcal{P}$ via (27);
11：**while** not converged **do**
12： Update $\mathcal{S}$ via Eq. (33); Update $\mathcal{Y}$ via Eq. (36); Update $\mathcal{T}$ via Eq. (37);
13： Update $\mathcal{M}$ via Eq. (39); Update multipliers via Eq. (41);
14：**end**
**Output：** static background $\mathcal{B}$+ moving object $\mathcal{M}$。

## 4. Experimental results and discussion

To verify the effectiveness of the proposed method, the method in this paper was compared with several advanced rain removal methods, namely, the tensor model method Jiang [22], the directional regularization-based method Sun [25] and the group sparsity-based method Wang [26]. In the experiments, the regularization parameters $\{\alpha_1, \alpha_2, \cdots, \alpha_5\}$were manually selected from {1,10,100,1000} and $\{\beta_1, \beta_2, \cdots, \beta_6\}$ were set to 50. The effectiveness of these parameters was illustrated by comprehensive tests.

### 4.1 Synthetic rainy video experiment

Firstly, the artificially generated rain videos with different types of rain streaks are processed and compared. Three objective image quality metrics, peak signal-to-noise ratio (PSNR), structural similarity (SSIM) and residual (RES), can be used to quantitatively compare the rain removal effect. Fig.3 and Fig.4 show the original clean video of the video sequence "highway", the composite video of moderate rain and heavy rain, and the same frame of the video after rain removal by the proposed method, respectively. The images show that the proposed method can effectively remove the rain streaks and retain the detail information of the images. Table 1 shows the numerical results after different rain removal methods, and in most cases, the proposed method achieves better metrics.

Table 1 Quantitative comparisons of rain streaks removal on synthetic "Highway" video

| Highway | Method | PSNR($\mathcal{B}$) | SSIM($\mathcal{B}$) | SSIM($\mathcal{R}$) | RES($\mathcal{B}$) |
|---|---|---|---|---|---|
| Middle | Rainy | 29.566 | 0.578 | - | 50.480 |
| | Jiang et al.(a) | 33.954 | 0.736 | 0.525 | 31.014 |
| | Sun et al.(b) | 34.571 | 0.747 | 0.585 | 28.987 |
| | Wang et al.(c) | 40.177 | 0.867 | 0.880 | 16.246 |
| | Ours. (d) | 42.136 | 0.899 | 0.837 | 13.699 |
| Heavy | Rainy | 27.232 | 0.555 | - | 66.031 |
| | Jiang et al.(a) | 29.185 | 0.621 | 0.227 | 52.958 |
| | Sun et al.(b) | 31.151 | 0.704 | 0.478 | 42.347 |
| | Wang et al.(c) | 33.351 | 0.699 | 0.668 | 33.131 |
| | Ours. (d) | 34.748 | 0.711 | 0.759 | 31.265 |


Figure 3 Comparisons of deraining results on one frame extracted from the synthetic “highway” video under middle rainy conditions

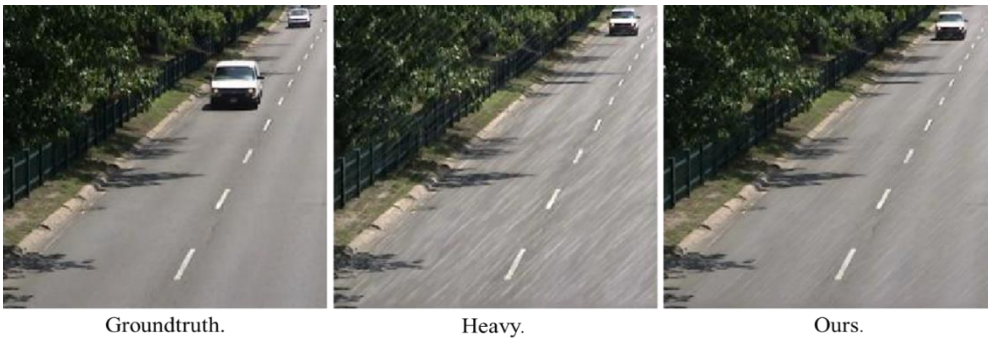


Figure 4 Comparisons of deraining results on one frame extracted from the synthetic “highway” video under heavy rainy conditions

Table 2 Quantitative comparisons of rain streaks removal on five different synthetic videos

| Video | Method | PSNR($\mathcal{B}$) | SSIM($\mathcal{B}$) | SSIM($\mathcal{R}$) | RES($\mathcal{B}$) |
|---|---|---|---|---|---|
| Park | Rainy | 31.698 | 0.373 | - | 36.796 |
| | Jiang et al.(a) | 36.305 | 0.561 | 0.517 | 21.704 |
| | Sun et al.(b) | 35.331 | 0.501 | 0.437 | 24.276 |
| | Wang et al.(c) | 42.359 | 0.747 | 0.874 | 10.921 |
| | Ours. (d) | 44.028 | 0.761 | 0.907 | 10.082 |
| Truck | Rainy | 31.195 | 0.739 | - | 46.857 |
| | Jiang et al.(a) | 37.118 | 0.881 | 0.689 | 25.791 |
| | Sun et al.(b) | 35.421 | 0.855 | 0.637 | 30.843 |
| | Wang et al.(c) | 39.528 | 0.958 | 0.802 | 20.091 |
| | Ours. (d) | 41.021 | 0.985 | 0.847 | 19.080 |
| Man | Rainy | 37.424 | 0.822 | - | 23.162 |
| | Jiang et al.(a) | 41.235 | 0.901 | 0.557 | 16.660 |
| | Sun et al.(b) | 40.776 | 0.971 | 0.793 | 17.701 |
| | Wang et al.(c) | 38.440 | 0.968 | 0.708 | 22.132 |
| | Ours. (d) | 42.726 | 0.982 | 0.811 | 15.876 |


Figure 5 Comparisons of deraining results on one frame extracted from the synthetic “man” video

Fig.5 shows the same frame from the original video of the synthesized video sequence "man", the synthesized rainy video streaks, and the video after the rain removal process. It can be seen that the method of Jiang et al [22] is not strong enough to remove the rain streaks, while the methods of Sun et al [25] and Wang [26] can effectively reduce the effect of rain streaks but tend to over-smooth the details. In comparison, the method in this chapter can be effective in removing rain streaks and can retain more background information. Table 2 shows the numerical results of three different sets of synthetic rain videos processed by different rain removal methods, and in most cases, the proposed method achieves better metrics.

### 4.2 Real rainy video experiment

Fig.6 shows three different frames extracted from the real video footage "Manhattan" after different rain removal methods. In contrast, the method proposed in this chapter is well adapted to this real scene and effectively removes the rain line while preserving the detail information.


Figure 6 Comparisons of deraining results on three frames extracted from the realistic "Manhattan" video

## 5. Conclusion

In this paper, for the interference of the moving objects in the video image with rain on the low-rank structure of the video background, the rainy video is trisected into static background, moving objects and rain streaks. After extracting the static background according to its low rank property, the dynamic objects are aligned and transformed using block matching to achieve low-rank processing, and then the low-rank sparse tensor model is solved by ADMM algorithm to complete the video rain streaks removal. The experimental results show that the method can effectively remove the rain streaks while preserving more details of the video background. In daily life, the method can be applied to real environments where rain removal is needed for video footage containing moving objects, such as outdoor people shooting on rainy days. The method in this paper still has two points that deserve improvement: (1) the ability to deal with rainstorms is insufficient; (2) the alignment transformation method for moving objects can be better optimized to improve the algorithm performance.

## References:

1. ROSER M, GEIGER A. Video-based raindrop detection for imp-roved image registration [C]// Proceedings of IEEE 12th International Conference on Computer Vision Workshops, Workshops, Kyoto, Japan, Sep, 2009: 570-577.
2. MAJI S, BERG A C, and MALIK J. Classification using intersection kernel support vector machines is efficient [C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, Jun, 2008: 1-8.
3. GARG K, NAYAR S K. Vision and Rain[J]. International Journal of Computer Vision. 2007, 75 (1): 3-27.
4. COMANICIU D, RAMESH V, MEER P. Kernel-based object tracking [J]. Pattern Analysis & Machine Intelligence. 2003, 25(5): 564-577.
5. ZHU Q, SHAO L, HENG P A, et al. A novel rain detection and removal approach using guided filtering and formation modeling [C]//Proceedings of the IEEE International Conference on Robo-tics and Biomimetics, Shenzhen, 2013: 563-567.
6. WANG C, SHEN M, YAO C. Rain streak removal by multi-frame-based anisotropic filtering [J]. Multimedia Tools & Application. 2017, 76(2): 2019-2038.
7. BARNUM P, KANADE T, NARASIMHAN S G. Spatio temporal frequency analysis for removing rain and snow from videos [C]//Proceedings of Workshop on Photometric Analysis for Computer Vision (PACV), in Conjunction with International Conference on Computer Vision, New York, 2007: 1-8.

8. SANTHASEELAN V, ASARI V K. Utilizing local phase information to remove rain from video [J]. International Journal of Computer Vision. 2015, 112(1): 71-89.
9. CHEN Z, SHEN J H. A new algorithm of rain (snow) removal in video [J]. Journal of Multimedia. 2013, 8(2): 168-179.
10. FADILI M J, STARCK J L, BOBIN J, MOUDDEN Y. Image decomposition and separation using sparse representations: an overview [J]. Proceedings of the IEEE. 2010, 98(6): 983-994.
11. KANG L W, LIN C W, LIN C T, et al. Self-learning-based rain streak removal for image/video [C]//Proceedings of IEEE International Symposium on Circuits and Systems, Seoul, Korea, 2012: 1871-1874.
12. SUN S H, FAN S P, WANG Y C F. Exploiting image structural similarity for single image rain removal [C]//Proceedings of IEEE International Conference on Image Processing, Paris, 2014: 4482-4486.
13. RAMYA C, RANI S S. Rain removal in image sequence using sparse coding [J]. Communications in Computer & Information ence, 2012, 330: 361-370.
14. ZHENG X, LIAO Y, GUO W, et al. Single-image-based rain and snow removal using multiguided filter [C]//Proceedings of 20th International Conference on Neural Information Processing, Part III, Lecture Notes in Computer Science 8228, Berlin, 2013: 258-265.
15. WANG Y, CHEN C, ZHU S, et al. A framework of single-image deraining method based on analysis of rain characteristics [C]//Proceedings of IEEE International Conference on Image Processing, IEEE, 2016: 4087-4091.
16. CHEN D, CHEN C, KANG L. Visual depth guided color image rain streaks removal using sparse coding [J]. IEEE Transactions on Circuits & Systems for Video Technology, 2014, 24(8): 1430-1455.
17. ZHOU Y, HAN Y S, ZHOU P C. A method for removing raindrops from a single image [J]. Journal of Graphics. 2015, 36(3): 438-443.
18. LI Y, TAN R T, GUO X, et al. Rain streak removal using layer priors [C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, Las Vegas, NV, USA, Jun, 2016: 2736-2744.
19. CHEN Y L, HSU C T. A generalized low-rank appearance model for spatio-temporally correlated rain streaks [C]//Proceedings of IEEE International Conference on Computer Vision. IEEE, 2013: 1968-1975.
20. KIM J H, SIM J Y, KIM C S. Video deraining and desnowing using temporal correlation and low-rank matrix completion [J]. IEEE Transactions on Image Processing, 2015, 24(9): 2658-2670.
21. OKATANI T, YOSHIDA T, DEGUCHI K. Efficient algorithm for low-rank matrix factorization with missing components and performance comparison of latest algorithms [C]//Proceedings of IEEE International Conference on Computer Vision. IEEE, 2011: 842-849.
22. JIANG T X, HUANG T Z, ZHAO X L, et al. A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors [J]. IEEE Conference on Computer Vison & Pattern Recognition (CVPR). Jul, 2017: 2818–2827.
23. LI M, XIE Q, ZHAO Q, et al. Video rain streak removal by multiscale convolutional sparse coding [C]//Proceedings of IEEE/CVF Conference on Computer Vison Pattern Recognition (CVPR), Jun, 2018: 6644–6653.
24. SUN Z, XIONG S, LIU R W. Directional regularized tensor modeling for video rain streaks removal [J]. 2019.
25. WANG Y T, ZHAO X L, JIANG T X, et al. A total variation and group sparsity based tensor optimization model for video rain streak removal[J]. Signal Processing Image Communication, 2018.
26. YANG W, TAN R T, FENG J, et al. Deep joint rain detection and removal from a single image [C]//Proceedings of IEEE CVPR, Honolulu, HI, USA, Jul, 2017: 1357-1366.
27. FU X, HUANG J, DING X, et al. Clearing the skies: A deep network architecture for single-image rain removal [J]. IEEE Transactions on Image Process. 2017, 26(6): 2944-2956.
28. LIU J, YANG W, YANG S, GUO Z. Erase or fill? Deep joint recurrent rain removal and reconstruction in videos [C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA. Jun, 2018: 3233-3242.
29. LIU J, YANG W, YANG S, GUO Z. D3R-Net: Dynamic routing residue recurrent network for video rain removal [J]. IEEE Transactions on Image Processing. 2019, 28(2): 699-712.

30. LIU R, JIANG Z, MA L, et al. Deep layer prior optimization for single image rain streaks removal [C]// Proceedings of IEEE ICASSP, Calgary, AB, Canada. Apr, 2018: 1408-1412.
31. RUDIN L I, OSHER S, FATEMI E. Nonlinear total variation based noise removal algorithms [J]. Physica D Nonlinear Phenomena. 1992, 60(1-4): 259–268.
32. CHAN S H, KHOSHABEH R, GIBSON K B, et al. An augmented Lagrangian method for total variation video restoration [J]. IEEE Transactions on Image Processing. 2011, 20(11): 3097–3111.
33. CHAMBOLLE A. An algorithm for total variation minimization and applications [J]. Journal of Mathematical Imaging and Vision. 2004, 20(1): 89–97.
34. CAO X, YANG L, GUO X. Total variation regularized RPCA for irregularly moving object detection under dynamic background [J]. IEEE Transactions on Cybernetics. 2016, 46(4): 1014-1027.
35. TAO M, YANG J, HE B. Alternating direction algorithms for total variation deconvolution in image reconstruction[J]. Optimization Online TR0918, Department of Mathematics, Nanjing University, 2009.
36. ZHOU X, YANG C, YU W. Moving object detection by detecting contiguous outliers in the low-rank representation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013, 35(3):597–610.
37. ORCHARD M T et al. Overlapped block motion compensation: an estimation-theoretic approach[J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 1994, 3(5):693–699.
38. ELAD M, AHARON M. Image denoising via sparse and redundant representations over learned dictionaries[J]. IEEE Tip, 2006, 15(12):3736–3745.
39. HUI J, LIU C, SHEN Z, et al. Robust video denoising using low rank matrix completion[C]// The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010. IEEE, 2010.
40. BARNEA D I, SILVERMAN H F. A class of algorithms for fast digital image registration[J]. IEEE Transactions on Computers, 2009, C-21(2):179-186.