Video Desnowing and Deraining Based on Matrix Decomposition

Weihong Ren[1,2,3], Jiandong Tian[1*], Zhi Han[1], Antoni Chan[3], Yandong Tang[1]
[1]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences,
[2]University of Chinese Academy of Sciences, [3]City University of Hong Kong
{renweihong, tianjd, hanzhi, ytang}@sia.cn, abchan@cityu.edu.hk

Abstract

The existing snow/rain removal methods often fail for heavy snow/rain and dynamic scene. One reason for the failure is due to the assumption that all the snowflakes/rain streaks are sparse in snow/rain scenes. The other is that the existing methods often can not differentiate moving objects and snowflakes/rain streaks. In this paper, we propose a model based on matrix decomposition for video desnowing and deraining to solve the problems mentioned above. We divide snowflakes/rain streaks into two categories: sparse ones and dense ones. With background fluctuations and optical flow information, the detection of moving objects and sparse snowflakes/rain streaks is formulated as a multi-label Markov Random Fields (MRFs). As for dense snowflakes/rain streaks, they are considered to obey Gaussian distribution. The snowflakes/rain streaks, including sparse ones and dense ones, in scene backgrounds are removed by low-rank representation of the backgrounds. Meanwhile, a group sparsity term in our model is designed to filter snow/rain pixels within the moving objects. Experimental results show that our proposed model performs better than the state-of-the-art methods for snow and rain removal.

1. Introduction

Raining and snowing have a negative effect on the visual quality and degrade the performance of various computer vision algorithms such as object detection [24], video tracking [2] and segmentation [28]. Thus, to make outdoor vision systems robust to different weather conditions, it is necessary to remove snow and rain in video sequences. In this paper, we use snow/rain to represent snowflakes/rain streaks for simplicity.

Recently, the research on snow/rain removal has attracted attention [14, 23, 30, 31, 32]. The existing snow/rain removal methods usually assume that all the snow/rain are sparse in a scene. Based on intensity fluctuations, falling directions or shapes, they first detect snow/rain.

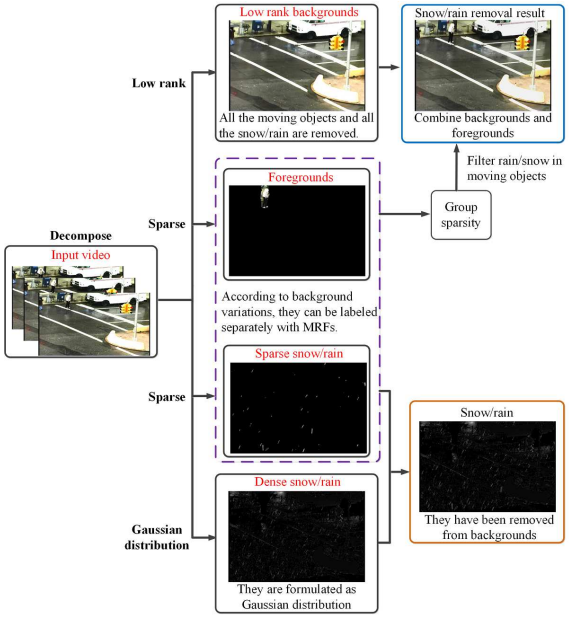


Figure 1: Composition of a general snow/rain scene: Low rank backgrounds, foregrounds, sparse snow/rain, and dense snow/rain. Sparse snow/rain usually have large intensity fluctuations, consistent directions and particular shapes, and they can be detected by the existing methods. However, dense snow/rain mislead the detection.

Then, snow/rain filters are applied to the corrupted pixels. Snow/rain are exactly sparse under light snow/rain and relatively static scene. However, heavy snow/rain scenes are not only occluded by sparse snow/rain, but also blurred by cluttered snow/rain with no obvious properties to be detected.

Fig. 1 shows the composition of a general rain scene: low rank backgrounds, foregrounds, sparse rain streaks, and dense rain streaks. Sparse rain streaks usually have large intensity fluctuations, consistent directions and particular shapes, and they can be detected by the existing methods. However, dense rain streaks, with no obvious properties, lead to wrong detection. This may be the reason why the existing methods do not perform well for heavy rain. In a rain scene, it is also difficult to separate moving objects

*Corresponding author

and sparse rain streaks. Thus, most of the existing methods often cause deformations and artifacts on moving objects.

For this issue, we propose a model based on matrix decomposition for video deraining or desnowing under heavy rain and dynamic scene (Section 3.1). We divide rain streaks into two categories: sparse ones and dense ones (see Fig.1). With background fluctuations and optical flow information, the detection of sparse rain streaks and moving objects is formulated as a multi-label MRFs (Section 3.2). As for dense rain streaks, they are assumed to obey a Gaussian distribution. All the rain streaks, including sparse ones and dense ones, in scene backgrounds are removed through low rank representations of the backgrounds. After the detection of moving objects, a group sparsity term is designed to filter rain pixels within the moving objects (Section 3.3). In this paper, there are mainly three contributions:

1. For heavy rain scenes, rain streaks have complex photometric and physical properties, and a single model is difficult to model all the rain streaks. Thus, we divide rain streaks into sparse ones and dense ones, and model them separately in a matrix decomposition framework. This process makes our model effective at tackling with heavy rain.
2. Misled by rain streaks, moving objects are difficult to be detected and filtered in a rain scene. Based on background fluctuations and flow information, we formulated the detection of sparse rain streaks and moving objects as a multi-label MRFs.
3. Due to wrong detections or improper filtering, the existing methods often cause deformations and artifacts on moving objects. To avoid this problem, we design a group sparsity term to filter rain pixels within moving objects.

2. Related work

A pioneering work on detecting and removing rain streaks in a video was proposed in [11], where its authors presented the comprehensive analysis of the visual effects of raindrops on an imaging system, and developed a correlation model that detects the dynamics of raindrops and a physics-based motion blur model that explains the photometry of rain. However, when a moving object is mixed with rain streaks, their method usually generates false detection between rain streaks and the moving object [17]. They further showed that camera parameters, such as exposure time and depth of field, can be selected so as to reduce or even remove the effects of rain without altering the appearance of rain [12]. By using temporal and chromatic property, Zhang et al. [36] proposed a K-means ($k = 2$) clustering method to detect candidate rain pixels. An intensity histogram of each pixel is firstly computed. Then, according to the two peaks of the histogram, the rain intensity distribution can be obtained. However, their method usually blurs image because of a temporal average of the background. Barnum et al. [1] demonstrated a method for globally detecting and removing rain and snow by using a physical and statistical model to suppress certain spatio-temporal frequencies. The disadvantage of this approach is that changes in the frequency domain do not always cause pleasing effect in the spatial domain, and frequency-based detection has errors when frequencies corresponding to rain are too cluttered [32]. Bossu et al. proposed selection rules based on photometry and size to select the potential rain streaks or snowflakes in [4]. However, their assumptions, such as uniform velocities and directions of rain streaks, limit the method's performance.

Recently, to deal with heavy rainfall in dynamic scenes, Chen et al. [6] proposed a deraining method by estimating moving objects based on optical flow and scene clustering. Though both spatial and temporal information is adaptively exploited during rain pixel recovery, their method often cannot handle scenes taken from moving cameras. Yao et al. [34] developed a Bayesian probabilistic approach to solve the rain streaks detection problem. Rain temporal motion is assumed to be Pathological Motion, but this method fails to detect rain streaks within moving objects. Asari et al. studied the characteristics of rain streaks in [30], and designed a rain removal framework based on Alpha-blending. Due to the difficulty of determining the blending factor, the reconstructed scene background sometimes has blurring artifacts. Kim et al. [19] proposed a video deraining and desnowing method based on temporal correlation and low-rank matrix completion. Their method can handle dynamic scenes, but it often causes deformation and artifacts for motion objects with large displacement. The reason may be that they do not explicitly detect the moving objects.

In other related work like [16], the authors proposed a single-image-based rain removal framework by formulating rain removal as an image decomposition problem based on a sparse representation. Also with sparse representation, Luo et al. [25] built up a nonlinear generative model of rain image to separate a rain layer and derained layer. The two methods are not effective for heavy and cluttered rain/snow due to their inexact models. Kim et al. [18] observed that a typical rain streak has an elongated elliptical shape with a vertical orientation. By analyzing the rotation angle and the aspect ratio of the elliptical kernel, they proposed a rain streak detection method for a single image. Based on Gaussian mixture models and patch priors, Li et al. [21] formulate rain streak removal in a single image as a layer decomposition problem, but their method may cause blur and oversmoothness. Chen et al. [7] assumed that rain streaks usually reveal similar and repeated patterns on imaging scene, and proposed a low-rank model from matrix to tensor structure in order to capture the spatio-temporally correlated rain streaks. However, rain streaks do not always have similar patterns for heavy rain. Eigen et al. [10] trained a special-

ized form of convolutional neural network to restore an image taken through a window covered with dirt or rain. You et al. [35] focused on modelling, detecting and removing raindrops adhered to a windscreen or window glass. The methods in [10, 35] are effective for raindrops removal adhered to window glass but ineffective for rain in a scene. In this paper, our method mainly remove rain and snow in a scene, and it might fail for adherent rain or dirt.

3. Our model

In this section, we introduce our rain removal model based on matrix decomposition, and explain how to use MRFs to detect sparse outliers (i.e. sparse rain streaks and moving objects). We also introduce how to use a group sparsity property to filter the rain pixels within moving objects.

3.1. General model

Let $I = [I_1, I_2, ..., I_n] \in R^{m\times n}$ be a video sequence with n frames, which is corrupted by rain or snow, and $I_j \in R^m$ denotes the jth frame of the video sequence. The ith pixel in the jth frame is denoted as I_{ij}. The input video I can be regarded as a combination of B, F, S_s and S_d, i.e.

$$I = B + F + S_s + S_d, \tag{1}$$

where $B \in R^{m\times n}$ is a matrix with the same size of I, which denotes the clear backgrounds; $F \in R^{m\times n}$ represent the intensity fluctuations caused by foregrounds; $S_s \in R^{m\times n}$ and $S_d \in R^{m\times n}$ denotes the intensity fluctuations caused by sparse rain streaks and dense rain streaks, respectively.

Backgrounds intensity should be unchanged, and thus they are linearly correlated with each other, which forms a low-rank matrix B. We impose the following constraint on B:

$$rank\,(B) \le \kappa,$$

where κ is a constant, which constrains the complexity of the background model.

The intensity fluctuations over video sequences are mainly caused by sparse outliers, i.e., moving objects F and sparse rain streaks S_s, and they are regarded as sparse matrices.

For S_d, we regard it as the fluctuations caused by dense rain streaks. Actually, it may contain fluctuations caused by background noise and illumination variations. In this paper, we simply assume it as Gaussian distribution,

$$P\,(S_d) = \frac{1}{\left(\sqrt{2\pi\sigma_d^2}\right)^N} \exp\left(-\frac{\|S_d\|_F^2}{2\sigma_d^2}\right), \tag{2}$$

where σ_d is the standard deviation of the Gaussian distribution, and N is the number of image pixels.

The foregrounds, namely moving objects, have similar structures in a video, and can be formulated via group sparsity. Referring to [5] [37], we detect and remove rain streaks through solving a matrix decomposition problem:

$$\begin{aligned}\min_{B,F,S_s,S_d} &\frac{1}{2{\sigma_d}^2}\|S_d\|_F^2 + \eta\cdot rank\,(B) + \lambda_1\|S_s+F\|_0 \\ &+ \|\mathcal{P}\,(F)\|_G, \\ s.t.\quad & I = B + F + S_s + S_d,\end{aligned} \tag{3}$$

where σ_d, η and λ_1 are regularization parameters. The proper choice of these parameters will be discussed in the following part. After obtaining intensity fluctuations F, the operator $\mathcal{P}$ first extracts foregrounds from I, and then it applies block matching on the foregrounds. The pseudo-matrix norm $\|\mathcal{P}\,(F)\|_G$ makes foregrounds group sparse after the operation of $\mathcal{P}$.

In order to deal with dynamic scenes taken by a moving camera, we first align neighboring frames to a target frame before the target frame is derained or desnowed. Let $I_j \circ \tau_j$ denote the frame after the transformation parameterized by vector τ_j. Then, the proposed decomposition (1) becomes $I\circ\tau = B + F + S_d + S_s$, and $I\circ\tau = [I_1\circ\tau_1, I_2\circ\tau_2, \ldots, I_n\circ\tau_n]$. We also use nuclear norm to replace the rank operator in (3) [29], and the final model can be written as:

$$\begin{aligned}\min_{B,F,S_s,\tau} &\frac{1}{2\sigma_d^2}\|I\circ\tau - B - F - S_s\|_F^2 + \eta\cdot\|B\|_* \\ &+ \lambda_1\|S_s+F\|_0 + \|\mathcal{P}\,(F)\|_G.\end{aligned} \tag{4}$$

3.2. Moving objects and sparse rain streaks modeling with MRFs

Moving objects in a rain scene are difficult to handle. If a frame is directly filtered without knowing the exact locations of moving objects in it, deformations and artifacts are usually caused on the moving objects (Fig. 5). To avoid this problem, in contrast to other methods [11, 19, 31, 36], we explicitly detect the moving objects with MRFs for group sparsity filtering.

Eq.(4) is intractable to optimize due to the existence of sparse outliers S_s and F. Let $M \in \{0,1\}^{m\times n}$ be a two-valued matrix denoting the sparse outliers support (S_s and F):

$$M_{ij} = \begin{cases} 0, \text{ if } I_{ij} \text{ is background,} \\ 1, \text{ if } I_{ij} \text{ is sparse outlier.}\end{cases} \tag{5}$$

We set $E = I\circ\tau - B$ to represent background fluctuations. As long as $E_{ij} \neq 0$, i.e. $M_{ij} \neq 0$, we must have $E_{ij} = F_{ij}$ or $E_{ij} = (S_s)_{ij}$ to minimize (4). Thus, (4) can be written as:

$$\min_{B,M}\frac{1}{2\sigma_d^2}\|(1-M)\odot E\|_F^2 + \lambda_1\|M\|_1 + \eta\cdot\|B\|_* + \|\mathcal{P}\,(F)\|_G, \tag{6}$$

where $\odot$ denotes the pixel-wise multiplication. When B and τ are fixed, the optimization for M is determined by λ_1 and the background variations E. Eq.(6) can be further written as:

$$\min_{M} \sum_{ij} \left(\lambda_1 - \frac{1}{2\sigma_d^2} (E_{ij})^2 \right) \cdot M_{ij}. \tag{7}$$

When dependency between neighbors is considered, (7) can be rewritten as a first-order MRFs:

$$\min_{M} \sum_{ij} \left(\lambda_1 - \tfrac{1}{2\sigma_d^2} (E_{ij})^2 \right) \cdot M_{ij} + \sum_{ij} \sum_{kl \in \mathcal{N}_{ij}} \gamma \cdot \psi(M_{ij} - M_{kl}), \tag{8}$$

where γ is a regularization parameter, $\mathcal{N}_{ij}$ is the set of 4-connected neighbors of M_{ij}, and

$$\psi(x) = \begin{cases} 0, \text{ if } x = 0, \\ 1, \text{ if } x \neq 0. \end{cases} \tag{9}$$

Through (8), backgrounds and sparse outliers are separated. However, it is still difficult to differentiate F and S_s.

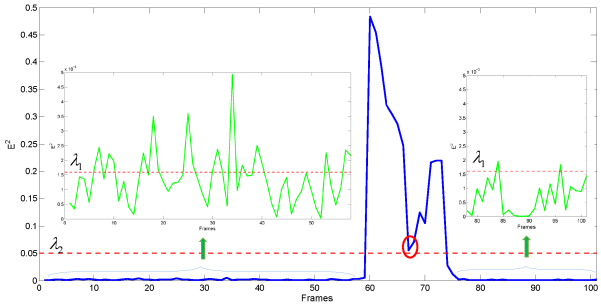


Figure 2: In both ends of the plot, intensity fluctuations in backgrounds vary irregularly due to rain, and there is a rough threshold λ_1 to distinguish rain pixels and backgrounds. Another threshold denoted as λ_2, caused by a moving object, is available to differentiate moving object and backgrounds.

Through analysis of many rain and snow video sequences, we find that intensity fluctuations caused by moving objects are more drastic than those caused by rain streaks or snowflakes. Fig.2 shows the plot of background fluctuations E in 100 consecutive frames for a pixel at particular position. In both ends of the plot, the intensity fluctuations in the backgrounds vary irregularly due to rain, and there is a threshold λ_1 to distinguish sparse outliers and the backgrounds. This analysis is reflected in (8). From frame 60 to frame 75, a moving object passes through backgrounds and causes great intensity shifts.

Another threshold denoted as λ_2 is available to differentiate the moving object and rain streaks, even though rain streaks, marked in a red circle in Fig.2, sometimes appear in the moving object and decrease the intensity shifts. In order to differentiate F and S_s, we rewrite M in (5) as:

$$\overline{M}_{ij} = \begin{cases} 0, \text{if } I_{ij} \text{ is background}, \\ 1, \text{if } I_{ij} \text{ is sparse rain/snow } S_s, \\ 2, \text{if } I_{ij} \text{is moving object } F. \end{cases} \tag{10}$$

Meanwhile, we also rewrite (8) as a multi-label MRFs:

$$\min_{\overline{M}} E_d + \sum_{ij} \sum_{kl \in \mathcal{N}_{ij}} \gamma \cdot \psi(\overline{M}_{ij} - \overline{M}_{kl}), \tag{11}$$

where

$$E_d = \begin{cases} \frac{1}{2\sigma_d^2} (E_{ij})^2, \text{ if } \overline{M}_{ij} = 0, \\ \left| \lambda_1 - \frac{1}{2\sigma_d^2} (E_{ij})^2 \right|, \text{ if } \overline{M}_{ij} = 1, \\ \left| \lambda_2 - \frac{1}{2\sigma_d^2} (E_{ij})^2 \right|, \text{ if } \overline{M}_{ij} = 2. \end{cases} \tag{12}$$

To make λ_1 robust to background changes such as illumination variation and video noise, we combine background fluctuations E with optical flow information. Based on the optical flow method [22], we first estimate the 2D flow field $\mathbf{u}(x, y)$ between two adjacent frames in a video. Then, we warp the next frame I_{k+1} to obtain $\bar{I}_k$, and calculate the difference between I_k and $\bar{I}_k$.

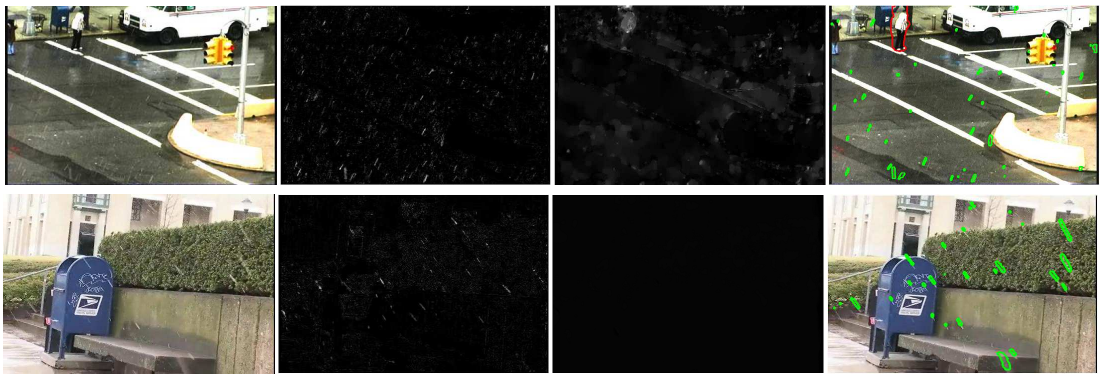
$$\begin{aligned} \bar{I}_k &= I_{k+1} \left(\mathbf{x} + \mathbf{u} \left(x, y \right) \right), \\ I_r &= \left| I_k - \bar{I}_k \right|. \end{aligned} \tag{13}$$

Since rain streaks are too small and move too fast to affect the optical flow estimation, we use I_r to guide MRFs for optimizing $\overline{M}$.

$$E_d = \begin{cases} \frac{1}{2\sigma_d^2} (E_{ij})^2, \text{ if } \overline{M}_{ij} = 0, \\ \left| \lambda_1 + \lambda_r - \frac{1}{2\sigma_d^2} (E_{ij})^2 - \omega_1 \left(I_r \right)_{ij}^2 \right|, \text{ if } \overline{M}_{ij} = 1, \\ \left| \lambda_2 - \frac{1}{2\sigma_d^2} (E_{ij})^2 \right|, \text{ if } \overline{M}_{ij} = 2, \end{cases} \tag{14}$$

where ω_1 controls the weight between background variations and optical flow information. The thresholds λ_1 and λ_2 are related to background variations E, and λ_r is related to optical information. The proper choice of these parameters will be discussed in the next section.

In Fig.3, rain streaks and moving objects can be accurately detected with Markov Random Fields. With the locations of moving objects, a group sparsity term is particularly designed to filter rain pixels within them. This process can effectively avoid deformations and artifacts of the moving objects.



Input image | Rain or snow map I_r | Motion map $\mathbf{u}$ | Sparse outliers

Figure 3: The detected sparse outliers with Markov Random Field. With local spatial information I_r and $\mathbf{u}$, moving objects and sparse rain streaks or snowflakes can be detected.

3.3. Group sparsity for moving objects

With local or non-local filtering algorithms, the existing methods usually remove rain within moving objects in a pixel-wise manner, and this process often causes blur and artifacts on the moving objects. Since image patches have non-local similarity in spatial and temporal space [9] [15], we design a group sparsity term to filter moving objects in a patch-wise manner.

Based on $\overline{M}$ obtained from (11), the operator $\mathcal{P}(F)$ first extracts foregrounds from I. Then, it divides foregrounds into overlapped patches and groups similar patches together within specific search windows centered on reference patches. For a target frame, similar patches are only searched in its two neighboring frames, i.e. three successive frames. For more details of patch-matching, please refer to [8]. The pseudo matrix norm $\|\cdot\|_G$ seeks for group sparsity of a patch set. After the patch set is processed, we put image patches back through a weighted average approach.

$(\mathcal{P}(F))_{i,j}$ is a patch set in $\mathcal{P}(F)$, and its reference patch is the jth patch in the ith frame. Let $Y = (\mathcal{P}(F))_{i,j}$ and $Y=[y_1, y_2, ..., y_l] \in R^{h\times l}$, where $y_1, y_2, ..., y_l$ are the similar pathes in foregrounds extracted from three successive frames. According to the group/structured sparsity [26] [38], the group sparsity term $\|Y\|_G$ can be solved by

$$(U, \Sigma, V) = \underset{U,\Sigma,V}{\arg\min} \left\|Y - U\Sigma V^T\right\|_F^2 + \mu\sum_{i=1}^{r}\sigma_i, \quad (15)$$

where $\Sigma = \operatorname{diag}\{\sigma_1, \sigma_2 ... \sigma_r\}\,(r = \operatorname{rank}(Y))$. Eq.(15) can be interpreted by singular value decomposition (SVD) from a bilateral variance estimation perspective [9]. Since rain streaks are randomly distributed in spatial and temporal field, the rain pixels within the moving objects can be removed by (15).

4. Algorithm

The objective function (4) is nonconvex, and is hard to obtain the solution in one step. Hence, we adopt an alternating algorithm that separates the energy minimization over B, τ, F and S_s. The step for B and τ is a convex problem, and the SOFT-INPUT algorithm [27] [29] has been proved effective for nuclear norm optimization problem. The step for S_s and F is solved by (11) to obtain $\overline{M}$. Based on $\overline{M}$, the operation $\mathcal{P}$ on F then filters rain within foregrounds by (15). The final rain removal result is obtained by combining clear backgrounds and filtered foregrounds. The overall algorithm is presented in Algorithm 1.

4.1. Reconstruct background B

When sparse outliers support M (i.e. $\overline{M}$) and transformation τ are fixed, according to (6), the minimization of objective function (4) can be transformed into:

$$\min_{B,\tau} \frac{1}{2\sigma_d^2}\left\|(1-M)\odot(I\circ\tau - B)\right\|_F^2 + \eta\cdot\|B\|_*. \quad (16)$$

Referring to [27], the solution to B in (16) can be given by SVD:

$$B{=}\operatorname{svd}\Big((1-M)\odot(I\circ\tau) + M\odot\tilde{B}\Big)_\alpha = U\Sigma_\alpha V^T, \quad (17)$$

where $\Sigma_\alpha = \operatorname{diag}\left[(d_1-\alpha)_+, ..., (d_r-\alpha)_+\right]$, $t_+ = \max(t, 0)$, and $\Sigma = \operatorname{diag}[d_1, ..., d_r]$. In the computation process, $\tilde{B}$ can be arbitrary initialized, and B is computed by iteratively using (17). The parameters η and σ_d^2 in (16) control the complexity of the scene background. A larger η gives a smoother scene background. In our algorithm, a rough estimation of rank $\kappa = \sqrt{n}$ is firstly given, and n denotes the number of frames that we select to reconstruct the scene background. In this paper, we set $n = 15$ for light rain scene and $n = 31$ for heavy rain. Then we start from a large η, and experimentally η is initialized to be the second largest singular value of I. The parameter σ_d^2 is set to 0.5 to estimate the dense rain streaks. After each iteration of (17), if $\operatorname{rank}(B) \le \kappa$, we reduce η by a factor $\theta_1 = 1/\sqrt{2}$ and repeat (17) until $\operatorname{rank}(B) > \kappa$.

In (16), τ=$[\tau_1, \tau_2, ..., \tau_n] \in R^{p\times n}$ are the parameterizations of all the transformations. When the change in τ is small, this constraint can be approximated by linearizing the current estimate of τ [29]. For $\Delta\tau \in R^{p\times n}$, let $I \circ (\tau + \Delta\tau) \approx I \circ \tau + J_\tau \Delta\tau$, where J_τ is the Jacobian matrix of the images. This linearization leads to a convex optimization problem over $\Delta\tau$, when B and M are fixed:

$$\begin{aligned} &\Delta\tau^* \leftarrow \arg\min_{\Delta\tau} \|(1-M) \odot (I \circ \tau - B + J_\tau \Delta\tau)\|_F^2 \\ &\tau \leftarrow \tau + \Delta\tau^*. \end{aligned} \tag{18}$$

4.2. Detect continuous outliers M (i.e. $\overline{M}$)

The parameter ω_1 in (14) controls the weight between background fluctuations and optical flow information, and λ_1, λ_2, λ_r are related to sparse outliers. Typically, we set $\omega_1 = 0.5$, because I_r provides much information for S_s detection. Background variation $\lambda_1 = 4.5\sigma^2$, and $\lambda_2 = 12.5\sigma^2$, where σ^2 is estimated online by the variance of $(I \circ \tau - B)_{ij}$. Similarly, $\lambda_r = 12.5{\sigma_r}^2$, where ${\sigma_r}^2$ is estimated by I_r. In Section 3.2, Eq.(4) is transformed into (11) for differentiating F and S_s. When B and τ are given, (11) is a standard form of MRFs with multiple labels, which can be exactly solved by [3] [20]. When $\overline{M}$ is obtained,

$$M = \begin{cases} 0, \text{ if } \overline{M} = 0, \\ 1, \text{ if } \overline{M} \neq 0. \end{cases} \tag{19}$$

Algorithm 1 Overall Algorithm for Rain or Snow Removal

Input: Rain or Snow video sequence $I = [I_1, I_2, ..., I_n] \in R^{m\times n}$

Output: Rain or Snow removal sequence $I^* \leftarrow [I_1^*, I_2^* ... I_n^*]$

Initialize: Transformation τ, backgrounds B, Support $\overline{M}$

1. **Repeat**
2. Update τ through (18)
3. Reconstruct B through (17)
4. Estimate λ_1, λ_2, and λ_r in (14)
5. Optimize $\overline{M}$ through (11) with MRFs
6. **Until** convergence
7. Filter rain/snow within Foregrounds through (15)
8. Combine Foregrounds and backgrounds to obtain I^*

4.3. Filter rain pixels within moving objects

In Section 4.1, rain streaks in scene backgrounds are removed by low-rank representation of the backgrounds. The key problem is to filter the rain pixels within moving objects. Based on sparse outliers $\overline{M}$ obtained in Section 4.2, (15) employs three successive frames to seek for group sparsity over foreground patches, and it is regarded as a standard matrix factorization problem. Since rain streaks are randomly distributed in spatial and temporal field, the rain pixels within the moving objects can be removed by (15). The technique proposed by [9] is adopted to solve (15), and the parameter μ is generally set to 5.

5. Experiments

In order to evaluate the performance of our proposed model, various rain and snow video sequences are employed. They contain illumination variations, camera motions, moving objects, etc.

Fig.4 is a dynamic snow scene taken from a moving camera. The method of Garg and Nayar [11] can detect candidate snow pixels through computing the direction and correlation of snowflakes, and removes most of the detected snowflakes. But the background and the mailbox are blurred because of the average of neighboring pixels. For dynamic scene taken by a moving camera, the method of Zhang et al. [36] performs video stabilization by warping every frame to align with the first frame, but this stabilization technique leads to blur and over-smoothness in the scene. Due to no pre-alignment for moving video, the method of Tripathi et al. [31] misleads the rain detection, and the edges in the mailbox are severely degraded. The method of Kim et al. [19] can handle video taken from a moving camera well, but it can not remove all the snowflakes in the scene. We align neighboring frames to a target frame in advance before the target frame is desnowed or de-rained, and our method can remove almost all the snowflakes.

In Fig.5, we show a dynamic rain scene. There are several moving objects in the rain scene with different velocities and directions. For the fast moving object namely the car, the method of Tripathi et al. and ours can handle well, but other methods degrade edges of the moving object severely. The disadvantage in the method of Tripathi et al. [31] is that snowflakes can not be removed clearly in the background. With a low rank representation of the background, our model can remove almost all the snowflakes. Meanwhile, a group sparsity model is designed to filter rain streaks within moving objects, and it avoids deformation and artifacts of the moving objects in restored frame.

Fig.6 is a heavy rain scene taken with a smart phone across a window, and it is challenging to remove all the rain streaks. Since the rain is too heavy, all the methods cause blur on the background. However, our method can remove almost all the rain streaks with 31 frames to reconstruct scene backgrounds and detect moving objects. Similar to our method, Kim et al. [19] employ 5 frames to remove rain with low rank matrix completion, but the heavy rain limits their method's performance. The comparison seems unfair for Kim's method, but this is not the case. Kim's method doesn't use a explicitly way to detect moving objects, and it will cause blur and artifacts on moving objets if more than 5 frames are employed.

Fig.7 shows rain removal results of two synthetic rain scenes: *Street1* and *Street2*. The rain videos are synthesized using the technique in [13], and we just show the results of Kim [19] and ours since the PSNR values of other

Figure 4: Comparison of different snow/rain removal methods for a snow scene taken form a moving camera. We align neighboring frames to a target frame in advance before the target frame is de-snowed or de-rained. Thus, our model can detect almost all the sparse snowflakes with Markov Random Field, and it does not depend on directions and sizes of snowflakes.

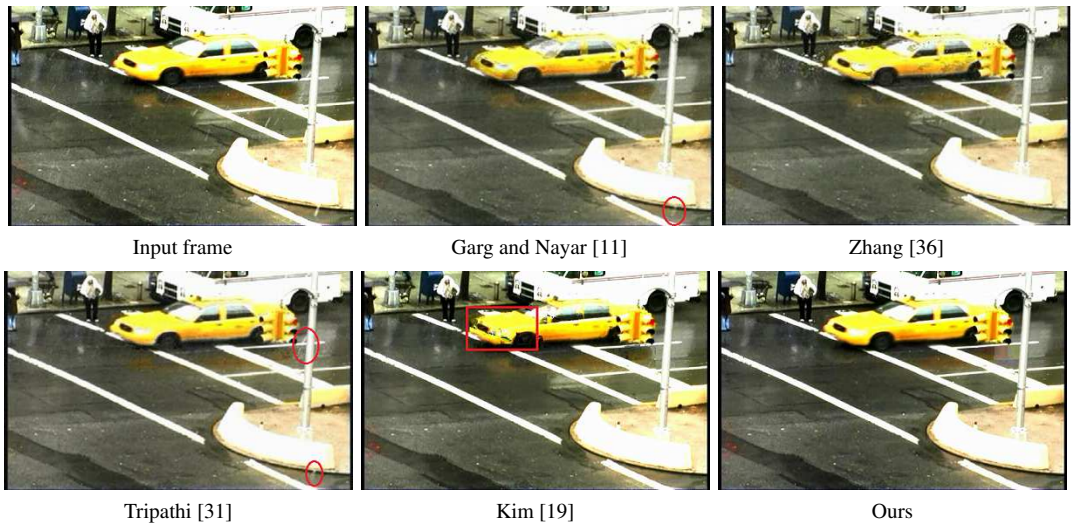


Figure 5: The rain removal results of a dynamic scene with different snow/rain removal methods. For the fast moving object, all the methods except ours, degrade the edges of moving object at different degrees. In our model, the moving object is filtered using group sparsity, and it avoids deformation and artifacts in restored frames.

methods are relatively lower (Tab.1). Though Kim's method can remove most of the rain streaks, it fails to handle wide and bright rain streaks. Our method divide rain streaks into sparse ones and dense ones, and all the snowflakes can be detected and removed through our model. The video comparison results including *Human4*, *Bike* and *Campus* can be found in *http://vision.sia.cn/our%20team/RenWeihong-homepage/vision-renweihong%28English%29.html*.

Tab.1 shows the quantitative comparisons for five synthetic rain videos using average PSNR value of a video. *Street1*, *Street2* and *Campus* are taken with a moving smart phone, and other two videos are chosen from [33]. Our method performs the best from the perspective of PSNR values, the reason is that our method can remove all the rain streaks. Due to improper filtering algorithms, the methods in [11, 31, 36] perform relatively poor for the synthetic heavy rain scenes. Kim's method in [19] achieves high PSNR values, but it fails to remove all the rain streaks.

Our Matlab code is carried on a PC with an Intel i5 CPU and 10G memory. For a 720×480 video frame, the running time of the provided Matlab code by Kim et al. is 126.4s, and ours is 136.5s. To our knowledge, few methods can perform well for snow/rain removal in real time. Zhang et al. uses the entire video to detect snow/rain. Tripathi and Garg

Figure 6: Comparison of different snow/rain removal methods under a heavy rain. Since the rain is too heavy, all the methods cause blur on the background. However, our method can remove almost all the rain streaks.

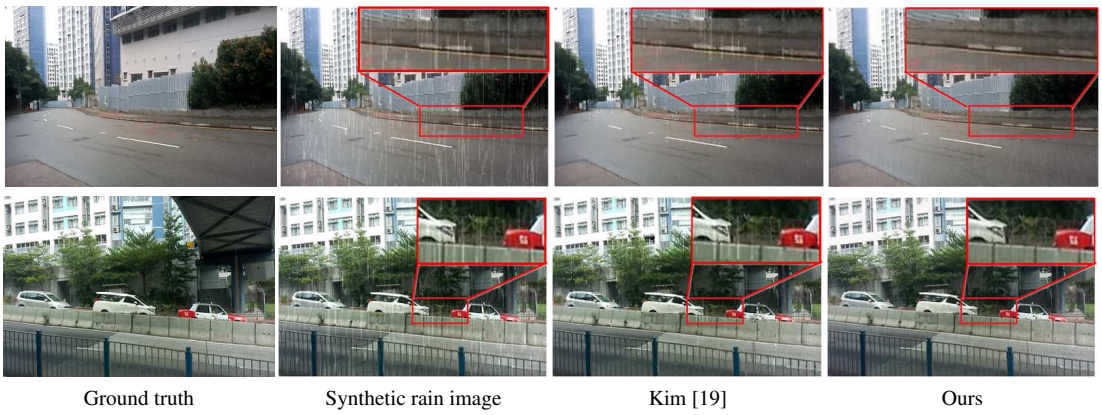


Figure 7: Comparison of two synthetic rain videos. Kim's method [19] can remove most of the rain streaks, but it fails to remove wide and bright rain streaks. Our method can handle all the rain streaks well with different models for rain streaks.

use 15 and 30 frames respectively to remove snow/rain.

Table 1: PSNR comparisons for five synthetic rain scenes

Scene	Rain video	Garg [11]	Zhang [36]	Tripathi [31]	Kim [19]	Ours
Street1	29.37	28.61	24.25	26.49	30.96	**31.40**
Street2	28.80	29.00	24.27	25.71	29.03	**29.62**
Campus	29.13	29.18	24.01	27.47	29.82	**30.45**
Human4	30.17	28.62	24.33	23.00	30.79	**31.35**
Bike	29.95	33.05	28.93	30.40	33.21	**33.54**

6. Limitation and Discussion

Currently, our method can remove all the rain streaks in the scene, but it may impair dynamic textures such as waves, swaying trees and crowed streets. In other words, our method may fail to separate dynamic textures and moving objects. To decompose rain video into more layers, such as moving object, sparse rain, dense rain and dynamic texture, may be a good choice, but it needs to extract more features for segmentation. The processing of dynamic textures and complex scenes will be our research focus.

7. Conclusion

In this paper, we divide rain streaks into sparse ones and dense ones, and model them separately in a matrix decomposition framework. The division makes our model more effective at handling heavy rain than other methods. Misled by rain streaks, it is difficult to detect and filter moving objects. Furthermore, improper filtering algorithm will cause deformations and artifacts on the moving objects. To avoid this problem, we first explicitly detect moving objects with MRFs, and then design a group sparsity term to filter rain pixels within them. The experiments on real and synthetic video sequences demonstrate that our method performs better than other state-of-the-art methods.

Acknowledgements

We thank Xi'ai Chen for the helpful discussions on experiments. The work was supported by the Natural Science Foundation of China under Grant Nos. 61473280, 61333019, 91648118 and 61303168. We also thank the support by Youth Innovation Promotion Association CAS.

References

[1] P. C. Barnum, S. Narasimhan, and T. Kanade. Analysis of rain and snow in frequency space. *International Journal of Computer Vision*, 86(2-3):256–274, 2010.

[2] J. Black, T. Ellis, and P. Rosin. A novel method for video tracking performance evaluation. *Proceedings of the IEEE InternationalWorkshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS 03)*, pages 125–132, 2003.

[3] A. Blake, P. Kohli, and C. Rother. *Markov random fields for vision and image processing*. Mit Press, 2011.

[4] J. Bossu, N. Hautière, and J.-P. Tarel. Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *International journal of computer vision*, 93(3):348–367, 2011.

[5] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[6] J. Chen and L.-P. Chau. A rain pixel recovery algorithm for videos with highly dynamic scenes. *Image Processing, IEEE Transactions on*, 23(3):1097–1104, 2014.

[7] Y. L. Chen and C. T. Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *2013 IEEE International Conference on Computer Vision (ICCV)*, pages 1968–1975, 2013.

[8] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *Image Processing, IEEE Transactions on*, 16(8):2080–2095, 2007.

[9] W. Dong, G. Shi, and X. Li. Nonlocal image restoration with bilateral variance estimation: a low-rank approach. *Image Processing, IEEE Transactions on*, 22(2):700–711, 2013.

[10] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *2013 IEEE International Conference on Computer Vision (ICCV)*, pages 633–640, 2013.

[11] K. Garg and S. K. Nayar. Detection and removal of rain from videos. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–528. IEEE, 2004.

[12] K. Garg and S. K. Nayar. When does a camera see rain? In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1067–1074. IEEE, 2005.

[13] K. Garg and S. K. Nayar. Photorealistic rendering of rain streaks. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 996–1002. ACM, 2006.

[14] K. Garg and S. K. Nayar. Vision and rain. *International Journal of Computer Vision*, 75(1):3–27, 2007.

[15] H. Ji, C. Liu, Z. Shen, and Y. Xu. Robust video denoising using low rank matrix completion. In *CVPR*, pages 1791–1798. Citeseer, 2010.

[16] L.-W. Kang, C.-W. Lin, and Y.-H. Fu. Automatic single-image-based rain streaks removal via image decomposition. *Image Processing, IEEE Transactions on*, 21(4):1742–1755, 2012.

[17] H. G. Kim, S. J. Seo, and B. C. Song. Multi-frame de-raining algorithm using a motion-compensated non-local mean filter for rainy video sequences. *Journal of Visual Communication and Image Representation*, 26:317–328, 2015.

[18] J. H. Kim, C. Lee, J. Y. Sim, and C. S. Kim. Single-image deraining using an adaptive nonlocal means filter. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 914–917, 2013.

[19] J. H. Kim, J. Y. Sim, and C. S. Kim. Video deraining and desnowing using temporal correlation and low-rank matrix completion. *Image Processing IEEE Transactions on*, 24:2658–2670, 2015.

[20] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):147–159, 2004.

[21] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Rain streak removal using layer priors. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[22] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):978–994, 2011.

[23] P. Liu, J. Xu, J. Liu, and X. Tang. Pixel based temporal analysis using chromatic property for removing rain from videos. *Computer and information science*, 2(1):p53, 2009.

[24] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[25] Y. Luo, Y. Xu, and H. Ji. Removing rain from a single image via discriminative sparse coding. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[26] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2272–2279. IEEE, 2009.

[27] R. Mazumder, T. Hastie, and R. Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322, 2010.

[28] D. W. Murray and B. F. Buxton. Scene segmentation from visual motion using global optimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2):220–228, 1987.

[29] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2233–2246, 2012.

[30] V. Santhaseelan and V. K. Asari. Utilizing local phase information to remove rain from video. *International Journal of Computer Vision*, 112(1):71–89, 2015.

[31] A. K. Tripathi and S. Mukhopadhyay. Video post processing: low-latency spatiotemporal approach for detection and removal of rain. *Iet Image Processing*, 6(2):181 – 196, 2012.

[32] A. K. Tripathi and S. Mukhopadhyay. Removal of rain from videos: a review. *Signal, Image and Video Processing*, 8(8):1421–1430, 2014.

[33] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2411–2418, 2013.

[34] C. Yao, C. Wang, L. Hong, and Y. Cheng. A bayesian probabilistic framework for rain detection. *Entropy*, 16(6):3302–3314, 2014.

[35] S. You, R. T. Tan, R. Kawakami, and K. Ikeuchi. Adherent raindrop detection and removal in video. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, 2013.

[36] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng. Rain removal in video by combining temporal and chromatic properties. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 461–464. IEEE, 2006.

[37] X. Zhou, C. Yang, and W. Yu. Moving object detection by detecting contiguous outliers in the low-rank representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(3):597–610, 2013.

[38] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *International Conference on Computer Vision*, pages 479–486, 2011.