sensors

MDPI

Article

Video Desnowing and Deraining via Saliency and Dual Adaptive Spatiotemporal Filtering

Yongji Li [1], Rui Wu [1], Zhenhong Jia [1,*], Jie Yang [2] and Nikola Kasabov [3]

[1] College of Information Science and Engineering, Xinjiang University, Urumqi 830046, China; liyongji@stu.xju.edu.cn (Y.L.); wurui@stu.xju.edu.cn (R.W.)
[2] Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200400, China; jieyang@sjtu.edu.cn
[3] Knowledge Engineering and Discovery Research Institute, Auckland University of Technology, Auckland 1020, New Zealand; nkasabov@aut.ac.nz
* Correspondence: jzhh@xju.edu.cn

Abstract: Outdoor vision sensing systems often struggle with poor weather conditions, such as snow and rain, which poses a great challenge to existing video desnowing and deraining methods. In this paper, we propose a novel video desnowing and deraining model that utilizes the salience information of moving objects to address this problem. First, we remove the snow and rain from the video by low-rank tensor decomposition, which makes full use of the spatial location information and the correlation between the three channels of the color video. Second, because existing algorithms often regard sparse snowflakes and rain streaks as moving objects, this paper injects salience information into moving object detection, which reduces the false alarms and missed alarms of moving objects. At the same time, feature point matching is used to mine the redundant information of moving objects in continuous frames, and a dual adaptive minimum filtering algorithm in the spatiotemporal domain is proposed by us to remove snow and rain in front of moving objects. Both qualitative and quantitative experimental results show that the proposed algorithm is more competitive than other state-of-the-art snow and rain removal methods.

Keywords: video desnowing and deraining; saliency; adaptive filtering; outdoor vision sensing

Citation: Li, Y.; Wu, R.; Jia, Z.; Yang, J.; Kasabov, N. Video Desnowing and Deraining via Saliency and Dual Adaptive Spatiotemporal Filtering. *Sensors* **2021**, *21*, 7610. https://doi.org/10.3390/s21227610

Academic Editor: Christophoros Nikou

Received: 30 October 2021
Accepted: 15 November 2021
Published: 16 November 2021

1. Introduction

Outdoor vision systems in traffic and safety applications have greatly promoted the development of society. Computer vision technologies, such as target tracking and human detection, are widely used. However, these technologies often confront challenges, such as heavy snow, rainstorms, strong winds and other poor weather conditions. Snowflakes and rain streaks can obscure key information in the video, and strong winds can shake the camera, which will make subsequent video processing more difficult. Therefore, removing snow and rain is an important part of computer vision.

In the early days, the photometric properties of rain were used to detect raindrops [1]. Some researchers utilized the direction and time attributes of rain streaks to remove rain [2,3]. However, the direction of snowfall is not consistent, so the direction attribute is not suitable for snow detection. Then, many researchers adopted filtering methods to remove snow and rain [2,4–7], but the cost of filtering is the loss of texture details in the background. Dictionary learning was adopted by some researchers to obtain rain dictionaries and non-rain dictionaries, but this method cannot completely remove rain [8].

Because the snowflakes and rain streaks in video do not cover the same pixels all the time, some researchers removed snow and rain [9–11] through the redundancy attributes between frames. However, the performance of this method is determined by selecting the

number of frames and background pixels. Although this method can remove most snow and rain, it easily leaves holes and artifacts on moving objects.

Recently, a patch-based Gaussian mixture model was used to reconstruct a clear background [12], and the Markov random field (MRF) was used to detect moving objects in videos [12–14], but through these methods, the edges of moving objects are distorted. Low rank is an important attribute of snow and rain video. Some researchers use low-rank matrix decomposition to remove snow and rain [13,15]. This method can accurately restore the background. However, robust principal component analysis (RPCA), adopted by Tian et al. [15], and MRF, adopted by Ren et al. [13], often fail to detect small moving objects because these two methods have difficulty distinguishing between moving objects and sparse snowflakes, often removing moving objects as snowflakes, or retaining sparse snowflakes as moving objects. Although these methods [12–14,16] can deal with snow or rain videos containing moving objects, none of them can effectively remove snow or rain from moving objects.

To solve the above problems, this paper utilizes low-rank tensor decomposition to remove snow and rain in the video, which makes full use of the spatial location information and the correlation between the three channels of the color video. This decomposition is more robust to heavy snow and rainstorm videos. Conventional moving object detection methods [17,18] have difficulty in distinguishing between sparse snowflakes and moving objects without rich textures. In this paper, saliency detection and moving object detection are combined to extract moving objects separately [19–21] because rain streaks and snowflakes have no salience information in video.

To effectively remove snow and rain from moving objects, we utilize the sparsity of snowflakes and rain streaks to remove snow and rain in front of the moving object for the first time through the combination of feature point matching and adaptive minimum filtering in the time domain. To achieve the best effect from removing snow and rain on moving objects of different sizes, we utilize adaptive minimum filtering in the spatial domain to obtain the final moving objects without snow and rain.

The main contributions of this paper are as follows:

- Due to the interference of rain streaks and snowflakes, the existing snow or rain removal algorithms cannot effectively detect moving objects. We introduce a saliency map into moving object detection, which improves the ability of moving object detection in snow and rain videos because almost all moving objects in snow and rain videos have salience information, while snowflakes and rain streaks do not.
- Because snow and rain in videos cannot cover the same pixels all the time, feature point matching is utilized by us to address the time continuity of moving objects in snow or rain videos and mine the redundant information of moving objects in continuous frames. A dual adaptive minimum filtering method in the spatiotemporal domain is proposed by us to remove snow and rain in front of moving objects.
- In contrast to matrix decomposition, our tensor decomposition makes full use of the spatial location information and the correlation between the three channels of the color video. In our decomposition, the background is relatively static, and we uniformly regard sparse and dense snowflakes, rain streaks and moving objects as sparse components.

The rest of this paper is organized as follows: Section 2 systematically introduces the main related work in removing snow and rain. Section 3 describes the proposed method. Section 4 presents the experimental analysis and results. Section 5 discusses the advantages and disadvantages of our proposed method. Our summaries and prospects are arranged in Section 6.

2. Related Work

We give a review on the methods of video snow and rain removal. The methods of single image snow and rain removal are also introduced for literature comprehensiveness.

2.1. Video Snow and Rain Removal Methods

Early researchers utilized the physical properties of snow and rain and the time attributes of frames to remove snow and rain. Garg et al. [1] first discussed the photometric characteristics of raindrops and developed a rain detection method based on a linear spatiotemporal correlation model. Zhang et al. [2] introduced chromaticity and time attributes to the intensity fluctuation of rain pixels, and *k*-means clustering was utilized to distinguish background and rain in videos. However, these methods are not suitable for rain videos with moving objects.

Later, some researchers utilized filtering methods to remove snow and rain. Park et al. [4] adopted the Kalman filter to remove rain. Shen et al. [9] combined a saturation filter, difference filter and white filter to detect snow particles. However, these methods lose the texture details of the image.

The temporal correlation of frames was used by some researchers to remove snow and rain. Based on the frame difference method, Huiying et al. [10] added the constraints of area and bearing to improve the accuracy of snow detection. Yang et al. [11] combined the frame difference method and L0 gradient minimization to remove snow. Brewer et al. [3] proposed a method to distinguish between rain and moving objects based on the shape and angle of the rain streaks, but it is difficult to remove heavy rain with it. Barnum et al. [22] believed that snow and rain in videos obey blurred Gaussian distributions, but the robustness of this method is poor. Bossu et al. [23] detected snow and rain through selection rules based on photometry and size. However, when the directions of snow and rain are not consistent, the result is not ideal.

Recently, some researchers have considered the time correlation of snow and rain video frames. Kim et al. [16] took into account global motion, local motion and snowflakes of various sizes in their snow removal algorithm. First, snowflakes are detected by the correlation of frames, and then snowflakes and outliers are distinguished by sparse representation and support vector machine (SVM). Finally, low-rank matrix completion is utilized to reconstruct the video sequence. However, this method cannot effectively remove heavy snow because only the correlation of five frames is taken into account. Ren et al. [13] proposed an algorithm to remove snowflakes or rain streaks based on matrix decomposition, which distinguishes moving objects from sparse snowflakes by setting threshold values for the pixel intensity at specific locations in continuous frames. However, in some experiments, moving objects are often missed. Tian et al. [15] first obtained a clean background by global low-rank matrix decomposition. Then, block matching based on the average absolute difference and local low-rank decomposition were used to remove snow in front of moving objects. However, the complexity of this method is too high. It is difficult to extract the low-rank structure of moving objects, especially for non-rigid motion.

In addition, Islam et al. [24] proposed a hybrid technique, where physical features and data-driven features of rain are combined to remove rain streaks in videos. Jiang et al. [25,26] used the sparsity of rain streaks to remove rain in videos. Similarly, Li et al. [14] proposed online multiscale convolutional sparse coding (MS-CSC) to remove snow and rain and adopted the MRF to detect moving objects. An affine transformation operation was utilized to update the background. In contrast to the previous MS-CSC model [27] designed for rain removal in a prefixed length of video, this method adjusts the parameters according to the correlation between previous and current frames to cope with streaming videos with continuously increasing frames in real time. Wei et al. [12] proposed a patch-based Gaussian mixture model, which uses MRF to distinguish moving objects from rain. On this basis, Yi et al. [28] proposed an online patch-based Gaussian mixture rain removal model, which can learn parameters adaptively.

2.2. Single Image Snow and Rain Removal Methods

Many researchers are working on using a single image to remove snow and rain. To make the related work more comprehensive, we also introduce snow and rain removal methods for a single image. Guided filtering is the main method of rain and snow removal for a single image [5–7,29]. However, guided filtering loses the details of the image when removing snow and rain, which makes this method not suitable for images with rich textures. Wang et al. [8] proposed an image decomposition method based on dictionary learning and guided filtering to obtain a clean background by removing the imagery layer where the snow and rain components are located. Unfortunately, this method still blurs the texture details of the background.

Deep learning is widely used to remove snow and rain from a single image. Qian et al. [30] injected the attention mechanism into the generation and discrimination network to improve the rain removability of the network. However, this method may not always be effective in removing rain streaks in complex scenes. Ren et al. [31] utilized a multistream DenseNet to estimate the rain location map, a generative adversarial network to remove the rain streaks and a refinement network to refine the details. Chen et al. [32] proposed a snow removal algorithm based on the snow size and a transparency-aware filter consisting of a snow size recognizer and a snow removal system that can identify transparency. A transparency-aware module removes snow with different scales and transparency, and a modified partial convolution algorithm removes nontransparent snow. However, the background is easily distorted in the actual performance.

In addition, Jaw et al. [33] used a pyramidal hierarchical design with lateral connections across different resolutions. The high-level semantic features were combined with other feature maps at different scales to enrich the location information. Liu et al. [34] proposed a multistage snow removal network. The network is mainly composed of translucency recovery (TR) and residual generation (RG) modules. The former is used to restore the background obscured by translucent snow particles. The latter generates an area obscured by opaque snow particles via the unoccluded area and the recovered area of the former. Li et al. [35] designed a multiscale stacked densely connected convolutional network (MS-SDN) to detect and remove snow. The network consists of a multiscale convolution subnet for extracting feature maps and two stacked modified DenseNets for snow detection and removal.

The main differences between our approach and previous methods are as follows:

- The previous desnowing and deraining algorithm cannot distinguish between sparse snowflakes/rain streaks and moving objects in heavy snow/rainstorms. We utilize saliency map to guide moving object detection, which can effectively avoid the influence of snowflakes/rain streaks.
- The existing desnowing and deraining algorithms cannot effectively remove the snowflakes and rain streaks in front of the moving object. Additionally, some methods deform the moving object. To solve these problems, we combine feature point matching and dual adaptive spatiotemporal filtering, proposed by us, to remove snowflakes and rain streaks in front of moving objects.

3. Proposed Method

In this section, we regard the snow video as a tensor, remove snow in the video by low-rank tensor decomposition, and then combine the saliency map with moving object detection to eliminate the interference of sparse snow, while extracting accurate moving objects. Finally, we utilize feature point matching and dual adaptive spatiotemporal filtering to remove the snow in front of the moving objects. The flow diagram of our proposed algorithm is shown in Figure 1.

3.1. Snow Video Background Modeling

The previous model converts snow video into the form of a matrix and then decomposes it into low-rank and sparse components. This decomposition can obtain a relatively clean background, but a major disadvantage of the matrix decomposition is that it can only deal with bidirectional (matrix) data, and the color snow/rain video data are a tensor.

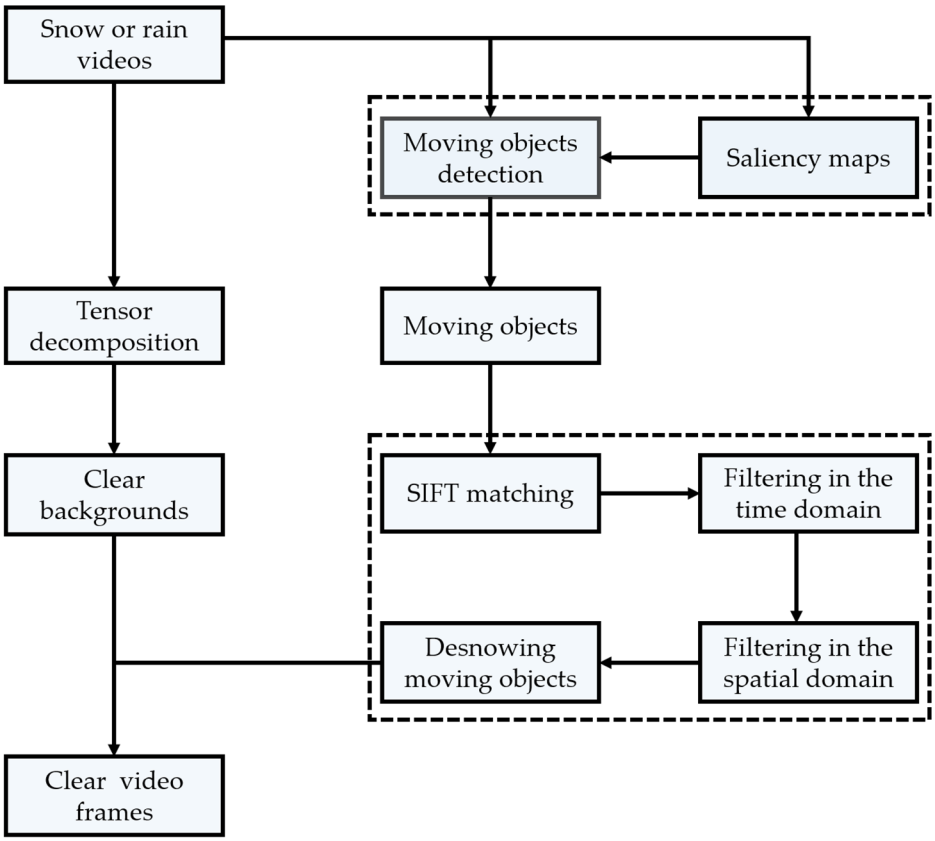


Figure 1. The flow diagram of our proposed algorithm.

The color frame is composed of three interrelated RGB channels. The matrix decomposition only deals with the three channels separately, which cannot make full use of the spatial location information, and the correlation between the three channels of the color video. This operation not only destroys the inherent structure of the original tensor, but also increases the computational cost of data analysis.

The natural advantage of the tensor is that one more dimension than the matrix can be used to store the RGB three-channel data. When decomposing the tensor, defining the tensor rank is an important problem. Unlike the rank of a matrix, researchers have many different definitions of the tensor rank, such as the CANDECOMP/PARAFAC (CP) rank [36], the Tucker rank [37], the tensor train (TT) rank [38], the tensor ring (TR) rank [39], and the tensor tubal rank [40]. In the restoration of color images and videos, the tensor tubal rank model based on the tensor–tensor product and tensor singular value decomposition (t-SVD) shows better performance than other rank models. The definitions of the tensor–tensor product, tensor singular value decomposition (t-SVD), tensor tubal rank, and tensor nuclear norm can be found in [41,42].

For a video sequence with a frame size of $h \times w$ and k frames, we consider a three-dimensional tensor $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, where $\mathbb{R}$ denotes the real number field. More precisely, the snow video is reshaped into a three-dimensional tensor $3 \times (hw) \times k$. Throughout this paper, we denote tensors by boldface Euler script letters. Our model is described as follows:

$$\begin{aligned} &\min_{\mathcal{L},\mathcal{S}} \|\mathcal{L}\|_* + \lambda\|\mathcal{S}\|_1 \\ &\text{s.t. } \mathcal{M} = \mathcal{L} + \mathcal{S}, \end{aligned} \tag{1}$$

where $\mathcal{M}$ is the reshaped input video, $\mathcal{M} \in \mathbb{R}^{3\times(hw)\times k}$, $\mathcal{L}$ is the low-rank background, $\mathcal{S}$ is the sparse component, $\|\bullet\|_*$ denotes the nuclear norm, and $\|\bullet\|_1$ denotes the $\ell_1-\text{norm}$. Generally, $\lambda = 1/\sqrt{(hw)k}$.

Then, the augmented Lagrangian function of (1) is as follows:

$$L(\mathcal{L}, \mathcal{S}, \mathbf{\Lambda}, \beta) = \|\mathcal{L}\|_* + \lambda\|\mathcal{S}\|_1 + \langle \mathbf{\Lambda}, \mathcal{L} + \mathcal{S} - \mathcal{M}\rangle + \frac{\beta}{2}\|\mathcal{L} + \mathcal{S} - \mathcal{M}\|_F^2, \tag{2}$$

where $\mathbf{\Lambda}$ is the Lagrange multiplier, β is the penalty parameter, $\langle\bullet\rangle$ denotes the inner product, and $\|\bullet\|_F^2$ denotes the square of the Frobenius norm.

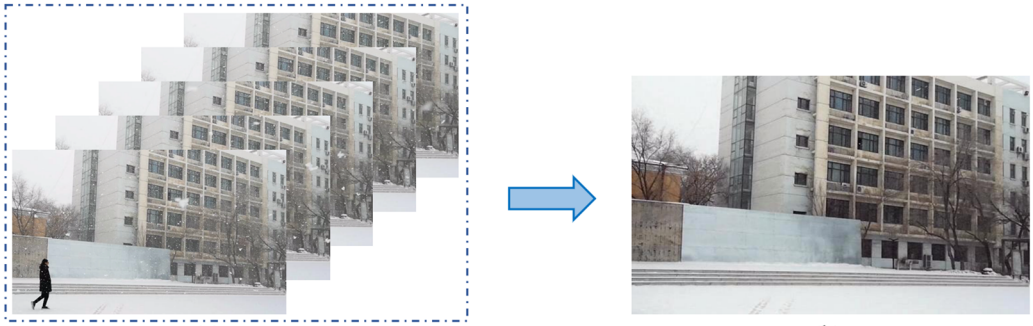
We iteratively solve the optimization problem through the framework of the alternating direction method of multipliers (ADMM) algorithm:

$$\begin{cases} \mathcal{L}_{k+1} = \arg\min_{\mathcal{L}} \|\mathcal{L}\|_* + \frac{\beta_k}{2}\left\|\mathcal{L} + \mathcal{S}_k - \mathcal{M} + \frac{\mathbf{\Lambda}_k}{\beta_k}\right\|_F^2 \\ \mathcal{S}_{k+1} = \arg\min_{\mathcal{S}} \lambda\|\mathcal{S}\|_1 + \frac{\beta_k}{2}\left\|\mathcal{L}_{k+1} + \mathcal{S} - \mathcal{M} + \frac{\mathbf{\Lambda}_k}{\beta_k}\right\|_F^2, \\ \mathbf{\Lambda}_{k+1} = \mathbf{\Lambda}_k + \beta_k(\mathcal{L}_{k+1} + \mathcal{S}_{k+1} - \mathcal{M}) \\ \beta_{k+1} = \min(\beta_k, \beta_{\max}) \end{cases} \tag{3}$$

The following equation serves as the stopping criterion for the above iterations:

$$\min\left\{\begin{matrix} \|\mathcal{L}_{k+1} + \mathcal{S}_{k+1} - \mathcal{M}\|_\infty, \\ \|\mathcal{L}_{k+1} - \mathcal{L}_k\|_\infty, \\ \|\mathcal{S}_{k+1} - \mathcal{S}_k\|_\infty \end{matrix}\right\} \le \varepsilon, \tag{4}$$

where ε is a very small number, e.g., 1×10^{-6}. Figure 2 shows the extracted low-rank component $\mathcal{L}$ of a snow video.



(a) (b)

Figure 2. Extracting the low-rank background (**b**) from a snow video sequence (**a**).

3.2. Moving Object Modeling

The conventional moving object detection methods have difficulty segmenting a complete moving object, and sparse snow is often recognized as a moving object, which results in snow that cannot be completely removed. To solve this problem, our proposed method combines the advantages of moving object detection and saliency detection, which introduces saliency items to form a new objective function. Specifically, we use a saliency map to guide moving object detection to strengthen the detectability of moving objects and weaken the impact of moving snow because snow tends to occupy most of the frame, which is not salient, while the moving object is salient. With the combination of a

saliency map and the motion detection, a complete moving object can be extracted separately.

In snow videos, moving objects without rich texture are prone to not being detected. To reduce false alarms and missed alarms, a saliency map is incorporated into an incremental subspace analysis framework, more accurate moving objects can be extracted. Our objective function systematically takes into account the properties of sparsity, low rank, connectivity, and saliency. The imposed saliency map avoids the interference of snow, and the connectivity plays a smooth role in the moving objects.

In the snow video, $\boldsymbol{c} \in \mathbb{R}^{N\times 1}$ denotes the current frame, where N is the number of pixels in the frame, i.e., $N = h \times w$. The goal is to find the locations of the moving objects in the current image $\boldsymbol{c}$. The moving object locations are represented by a foreground indicator vector $\overline{\boldsymbol{f}} \in \{0,1\}^N$, where 0 denotes the background and 1 denotes the foreground. The negative of the background indicator vector $\overline{\boldsymbol{b}}$ is identical to the foreground indicator vector $\overline{\boldsymbol{f}}$, i.e., $\overline{\boldsymbol{f}} = \boldsymbol{1} - \overline{\boldsymbol{b}}$, where $1 \in \mathbb{R}^{N\times 1}$, and the elements are all 1. $\overline{\boldsymbol{b}}$ is obtained by binarizing the background vector $\boldsymbol{b}$.

The background vector is obtained by the following minimization problem:

$$\min_{\boldsymbol{b},\boldsymbol{U},\boldsymbol{v}} \sum_{i=1}^{N} \begin{bmatrix} \frac{1}{2} b_i (\boldsymbol{U}_i \boldsymbol{v} - \boldsymbol{c}_i)^2 + \\ \beta(1-b_i) - \alpha b_i (1-s_i) \end{bmatrix} + \lambda \|\boldsymbol{D}\boldsymbol{b}\|_1, \tag{5}$$

where $\boldsymbol{U} \in \mathbb{R}^{N\times m}$ is a subspace matrix whose columns are orthonormal, m is the number of columns of $\boldsymbol{U}$, and $\boldsymbol{U}_i$ stands for the i th row of $\boldsymbol{U}$. The coefficient vector $\boldsymbol{v} \in \mathbb{R}^{m\times 1}$ is the low-dimensional representation of frame $\boldsymbol{c}$ in the subspace spanned by the rows of $\boldsymbol{U}$. $\boldsymbol{s} \in \mathbb{R}^{N\times 1}$ is the saliency map obtained by some salient object detection algorithms, such as those in [43–45], and s_i is the i th element of $\boldsymbol{s}$. $D = [D_h, D_v]^T$ is a difference matrix, and $\boldsymbol{D}_h$ and $\boldsymbol{D}_v$ are forward finite-difference operators in the horizontal and vertical directions, respectively. α, β and λ are the balancing parameters.

In Equation (5), $\boldsymbol{U}_i\boldsymbol{v}$ is the reconstruction of the background, and $\boldsymbol{U}_i\boldsymbol{v} - \boldsymbol{c}_i$ measures the similarity between $\boldsymbol{U}_i\boldsymbol{v}$ and $\boldsymbol{c}_i$. The second term $(1-b_i)$ makes the estimated foreground much sparser to avoid the interference of snow. The connectivity term $\|\boldsymbol{D}\boldsymbol{b}\|_1$ is minimized to smooth the foreground and background. Minimizing the object saliency term $-b_i(1-s_i)$ increases the chances that the foreground contains salient objects.

We utilize the alternating minimization method to seek the optimal variables $\boldsymbol{b}$, $\boldsymbol{U}$ and $\boldsymbol{v}$ in turn. It is extremely difficult to seek the optimal solution of $\boldsymbol{b}$ directly. We let $\boldsymbol{w} = \boldsymbol{b}$ and $\boldsymbol{h} = \boldsymbol{D}\boldsymbol{w}$. Equation (5) can be described as follows:

$$\begin{aligned} &\min_{\boldsymbol{b},\boldsymbol{U},\boldsymbol{v}} \sum_{i=1}^{N} \left[\tfrac{1}{2} b_i (\boldsymbol{U}_i \boldsymbol{v} - \boldsymbol{c}_i)^2 + \beta(1-b_i) - \alpha b_i (1-s_i) \right] + \lambda \|\boldsymbol{h}\|_1 \\ &\text{s.t. } \boldsymbol{w} = \boldsymbol{b}, \ \boldsymbol{h} = \boldsymbol{D}\boldsymbol{w}, \end{aligned} \tag{6}$$

With the Lagrange multiplier, the constraint term in Equation (6) is converted into the following unconstrained form:

$$\begin{aligned} &\min_{\mathbf{b},\mathbf{U},\mathbf{v},\mathbf{h},\mathbf{w}} \sum_{i=1}^{N} \left[\tfrac{1}{2} b_i (\boldsymbol{U}_i \boldsymbol{v} - \boldsymbol{c}_i)^2 + \beta(1-b_i) - \alpha b_i (1-s_i) \right] + \lambda \|\boldsymbol{h}\|_1 + \\ &\frac{\mu}{2} \|\boldsymbol{w} - \boldsymbol{b}\|_2^2 + \boldsymbol{x}^T (\boldsymbol{w} - \boldsymbol{b}) + \frac{\mu}{2} \|\boldsymbol{h} - \boldsymbol{D}\boldsymbol{w}\|_2^2 + \boldsymbol{y}^T (\boldsymbol{h} - \boldsymbol{D}\boldsymbol{w}), \end{aligned} \tag{7}$$

where $\mu/2\|\boldsymbol{w} - \boldsymbol{b}\|_2^2$ and $\boldsymbol{x}^T(\boldsymbol{w} - \boldsymbol{b})$ are obtained by converting $\boldsymbol{w} = \boldsymbol{b}$ into the unconstrained optimization function, and the vector $\boldsymbol{x}$ is the Lagrangian multiplier. Similarly,

$\mu/2\|\boldsymbol{h}-\boldsymbol{Dw}\|_2^2$ and $\boldsymbol{y}^T(\boldsymbol{h}-\boldsymbol{Dw})$ are obtained by converting the constraint $\boldsymbol{h}=\boldsymbol{Dw}$ into the unconstrained optimization function, and the vector $\boldsymbol{y}$ is the Lagrangian multiplier.

We solve the optimization problem (7) alternately to obtain the optimal variables.

We update $\boldsymbol{b}$ when $\boldsymbol{U,v,h,w,x}$ and $\boldsymbol{y}$ are fixed, as follows:

$$b_i = \frac{\beta + \mu\omega_i + x_i - \frac{1}{2}(\boldsymbol{U}_i\boldsymbol{v}-\boldsymbol{c}_i)^2 + \alpha(1-s_i)}{\mu}, \tag{8}$$

We update $\boldsymbol{h}$ when $\boldsymbol{b,U,v,w,x}$ and $\boldsymbol{y}$ are fixed, as follows:

$$\boldsymbol{h} = \arg\min_{\boldsymbol{h}} \frac{\lambda}{\mu}\|\boldsymbol{h}\|_1 + \frac{1}{2}\|\boldsymbol{h}-\boldsymbol{Dw}+\boldsymbol{y}/\mu\|_2^2, \tag{9}$$

The optimal solution is given by the following equation:

$$\boldsymbol{h} = S_{\lambda/\mu}(\boldsymbol{Dw} - \frac{\boldsymbol{y}}{\mu}), \tag{10}$$

We update $\boldsymbol{w}$ when $\boldsymbol{b,U,v,h,x}$ and $\boldsymbol{y}$ are fixed as follows:

$$\boldsymbol{w} = \arg\min_{\boldsymbol{w}} \frac{\mu}{2}\|\boldsymbol{w}-\boldsymbol{b}\|_2^2 + \boldsymbol{x}^{\mathrm{T}}(\boldsymbol{w}-\boldsymbol{b}) + \frac{\mu}{2}\|\boldsymbol{h}-\boldsymbol{Dw}\|_2^2 + \boldsymbol{y}^{\mathrm{T}}(\boldsymbol{h}-\boldsymbol{Dw}), \tag{11}$$

Equation (11) is a quadratic function of $\boldsymbol{w}$. Hence, the unique solution is the following:

$$\boldsymbol{w} = (\boldsymbol{I}+\boldsymbol{D}^T\boldsymbol{D})^{-1}\left[\boldsymbol{D}^T(\boldsymbol{h}+\frac{\boldsymbol{y}}{\mu}) + \boldsymbol{b} - \frac{\boldsymbol{x}}{\mu}\right], \tag{12}$$

We update $\boldsymbol{x}$ and $\boldsymbol{y}$ when $\boldsymbol{b,U,v,h}$ and $\boldsymbol{w}$ are fixed as follows:

$$\begin{cases} \boldsymbol{x}+\mu(\boldsymbol{w}-\boldsymbol{b}) \to \boldsymbol{x} \\ \boldsymbol{y}+\mu(\boldsymbol{h}-\boldsymbol{Dw}) \to \boldsymbol{y}, \\ d\mu \to \mu \end{cases} \tag{13}$$

where d is a parameter and its empirical value is 1.25.

We update $\boldsymbol{U}$ when $\boldsymbol{b,v,h,w,x}$ and $\boldsymbol{y}$ are fixed as follows:

$$\begin{aligned} &\boldsymbol{U} = \arg\min_{\boldsymbol{U}} \sum\nolimits_i \frac{1}{2}\boldsymbol{b}_i(\boldsymbol{U}_i\boldsymbol{v}-\boldsymbol{c}_i)^2 \\ &\text{s.t.} \quad \boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I}, \end{aligned} \tag{14}$$

where $\boldsymbol{I}$ is the identity matrix.

$\boldsymbol{v}$ is the low-dimensional representation of $\boldsymbol{c}$, which is given by the following:

$$\boldsymbol{v} = \boldsymbol{U}^T\boldsymbol{c} \tag{15}$$

3.3. Feature Point Matching and Dual Adaptive Spatiotemporal Filtering

In the adjacent frames, the change of the moving object is very small, but the snow moves very fast, which makes the feature point matching accurately match the moving object.

We utilize the scale invariant feature transform (SIFT) matching method to match moving objects in snow videos. The SIFT matching algorithm is robust to changes in object translation, brightness and scale. It includes five steps: (1) We construct scale space and detect extreme points to obtain scale invariance. (2) Unstable feature points are filtered for accurate positioning. (3) We extract feature descriptors from feature points and assign direction values to feature points. (4) Feature descriptors are utilized to find matching points. (5) The Euclidean distance of the feature vector is used as a similarity measure of

key points in two images. As shown in Figure 3, the SIFT matching we adopt can accurately match moving objects in different frames.

We paste the detected moving objects back into a low-rank background. When using feature point matching to remove snow in front of moving objects, one problem is that the number of matching frames directly determines the quality of snow removal in front of moving objects. To improve the robustness of the proposed method, we adaptively select the appropriate number of matching frames according to the speed of the moving object to strike a balance between over-smoothing and snow removal effects. General moving objects (such as pedestrians and cars) will produce unpleasant deformations in the spatiotemporal domain. If we measure the speed of the moving object according to the proportion of the coincident part of the moving object in the adjacent frame to the frame, there is a great error in the video with different resolutions. Therefore, we choose the proportion of the coincident part of the moving object in the adjacent frame to itself.

Each frame of the snow video reshapes a vector $\boldsymbol{c} \in \mathbb{R}^{N\times 1}$. We set the pixel coincidence rate between the moving object in the target frame $\boldsymbol{c}_o$ and the moving object in the previous frame $\boldsymbol{c}_{o-1}$ to χ_{o-1}. Similarly, the coincidence rate of the next frame is set to χ_{o+1}. When the coincidence rate is less than 80%, subsequent frames are no longer matched:

$$\boldsymbol{c}_{o+i} = \begin{cases} 0, & \text{if } \chi_{o+i} < 80\% \\ 1, & \text{if } \chi_{o+i} \geq 80\% \end{cases}, \mathrm{i} = \cdots, -2, -1, 1, 2, \cdots \tag{16}$$

where 0 indicates that the current frame $\boldsymbol{c}_o$ refuses to match $\boldsymbol{c}_{o+i}$, 1 indicates that the current frame $\boldsymbol{c}_o$ agrees to match $\boldsymbol{c}_{o+i}$.

If there are E and F frames matching $\boldsymbol{c}_o$ successfully forward and backward, respectively, then the reshaped matrix after matching is $\mathbb{R}^{N\times(E+F+1)}$. We select the smallest element value in each row as the result of time domain minimum filtering.

$$\boldsymbol{c}_{\tilde{o}} = \min[\boldsymbol{c}_{o-E}, \cdots, \boldsymbol{c}_{o-1}, \boldsymbol{c}_o, \boldsymbol{c}_{o+1}, \cdots, \boldsymbol{c}_{o+F}], \tag{17}$$

where $\boldsymbol{c}_{\tilde{o}}$ represents the result of adaptive minimum filtering in the time domain after the SIFT matching.

Because the time domain minimum filtering utilizes the correlation between frames, even if most of the moving object is covered by sparse snow, it can be recovered accurately.

In some cases, there are still unpleasant snow noises in the images after SIFT matching and minimum filtering in the time domain. To achieve a better snow removal effect, we introduce adaptive spatial domain minimum filtering:

$$\tilde{H}_{(i,j)} = \min\left\{ \begin{matrix} H_{(i-n,j-n)}, \cdots, H_{(i-n,j)}, \cdots, H_{(i-n,j+n)} \\ \vdots \qquad\quad \vdots \qquad\quad \vdots \\ H_{(i,j-n)}, \cdots, H_{(i,j)}, \cdots, H_{(i,j+n)} \\ \vdots \qquad\quad \vdots \qquad\quad \vdots \\ H_{(i+n,j-n)}, \cdots, H_{(i+n,j)}, \cdots, H_{(i+n,j+n)} \end{matrix} \right\}, \tag{18}$$

where H represents the pixel on the moving object, and i and j represent the horizontal and vertical coordinates of the target pixel, respectively. $\tilde{H}$ is the result of spatial domain minimum filtering. The size of the sliding window depends on the size of the moving object.

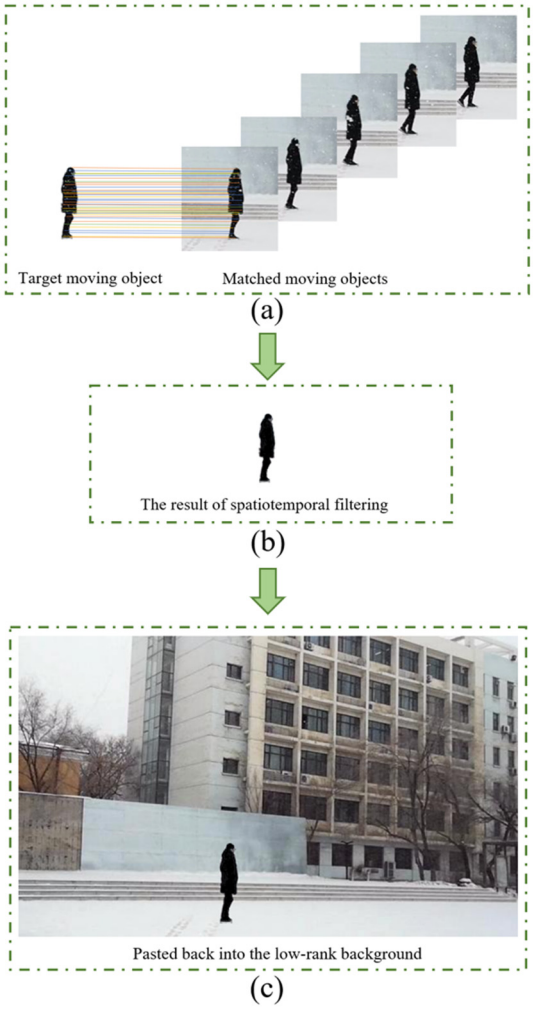


Figure 3. (**a**) The moving object matching processes, (**b**) the result of dual adaptive spatiotemporal filtering, (**c**) the clean video frame obtained by pasting the desnowing moving object back into a low-rank background.

4. Experiment

To show the superiority of our proposed method objectively and fairly, quantitative and qualitative evaluations are carried out in synthetic snow and rain videos, respectively. To further demonstrate the robustness of the proposed algorithm, the real snow and rain comparison scenes include heavy snow, rainstorms and dynamic background videos.

Our method is compared with state-of-the-art algorithms for removing snow and rain. The method of Kim et al. [16] was published in *Transactions on Image Processing* (TIP) in 2015 and not only effectively removes snow and rain, but it also has high robustness for dynamic scenes. The method of Wang et al. [8] was published in *Transactions on Image Processing* (TIP) in 2017, and it can remove snow and rain from a single image well. The algorithm of Li et al. [14] was published in *Transactions on Image Processing* (TIP) in 2021. Because this method updates parameters according to continuously increasing frames in real time, it can effectively remove snow and rain from dynamic scenes. The algorithm of Chen et al. [32] was presented at the *European Conference on Computer Vision* in 2020 and is currently the best snow removal method based on deep learning. All experiments were implemented on a PC with an i7 CPU and 32 GB RAM.

4.1. Comparation on Synthetic Snow and Rain Videos

We select two videos in CDNET database [46]. One of the scenes is called pedestrians, and the other is a challenging traffic intersection. Different degrees of snow and rain are added to the two videos. First, we qualitatively evaluate the snow and rain removal effect of the proposed algorithm and the four comparison algorithms. Then the quantitative evaluation results are given by comparing the peak signal-to-noise ratio (PSNR), the structural similarity (SSIM) [47], the feature similarity containing the chrominance information (FSIMc) [48] and the visual information fidelity (VIF) [49].

As shown in Figure 4, the method of Kim et al. [16] does not remove dense snowflakes and blurs the pedestrian's legs. There is still much snow in the results of Wang et al. [8] and Chen et al. [32]. Among the four comparison methods, the performance of the method of Li et al. [14] is the best, but there is still snow in the result. There is little snow in our result. In Figure 5, the method of Kim et al. [16] still blurs the moving car. The methods of Wang et al. [8] and Chen et al. [32] do little to remove rain. Similar to the result in Figure 4, there is still a little rain left in the result of Li et al. [14]. Our snow removal effect is still the best.

(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**)

Figure 4. Comparison on a synthetic snow video. (**a**) Ground truth, (**b**) input, (**c**) Kim et al. [16], (**d**) Wang et al. [8], (**e**) Li et al. [14], (**f**) Chen et al. [32], (**g**) proposed method.

(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**)

Figure 5. Comparison on a synthetic rain video. (**a**) Ground truth, (**b**) input, (**c**) Kim et al. [16], (**d**) Wang et al. [8], (**e**) Li et al. [14], (**f**) Chen et al. [32], (**g**) proposed method.

To compare the snow removal effects of the five methods more objectively, Tables 1 and 2 show the quantitative evaluation indices, such as PSNR, SSIM, FIMc and VIF, of each method. We calculate the average objective value of 200 frames of the above two videos. Our results are the best in every evaluation index, mainly because our proposed algorithm can effectively distinguish background from snow and rain. The method of Wang et al. [8] blurs the background, and the method of Chen et al. [32] distorts the background, which leads to the decline of their indices.

Table 1. Quantitative performance comparison of synthetic snow videos. All the results are the average of 200 frames.

Algorithm	Pedestrians			
	PSNR	SSIM	FSIMc	VIF
Kim et al. [16]	32.952	0.986	0.986	0.842
Wang et al. [8]	28.940	0.933	0.917	0.462
Li et al. [14]	35.395	0.987	0.988	0.832
Chen et al. [32]	25.963	0.898	0.916	0.506
proposed method	36.287	0.988	0.989	0.858

Table 2. Quantitative performance comparison of synthetic rain videos. All the results are the average of 200 frames.

Algorithm	twoPositionPTZCam			
	PSNR	SSIM	FSIMc	VIF
Kim et al. [16]	35.725	0.984	0.988	0.803
Wang et al. [8]	31.255	0.930	0.952	0.521
Li et al. [14]	37.848	0.982	0.984	0.795
Chen et al. [32]	23.698	0.837	0.895	0.501
proposed method	38.694	0.986	0.988	0.816

4.2. Comparation on Real Snow and Rain Videos

To further test the snow removability of our algorithm, in this section, we compare the proposed method with the four methods in real snow and rain videos.

Figure 6 shows a heavy snow scene, and Figure 7 shows a rainstorm scene. Although the methods of Li et al. [14] and Chen et al. [32] remove dense snow and rain, they cannot remove sparse snowflakes and rain streaks. The result of Kim et al. [16] is much better than the first two, but some snow and rain remain. Wang et al. [8] only removes dense snow and rain at the expense of image texture details. Because the main idea of this method is to remove snow and rain by filtering, the loss of background details is inevitable. In contrast, our method not only removes sparse and dense snow and rain, but it also restores a clear background.

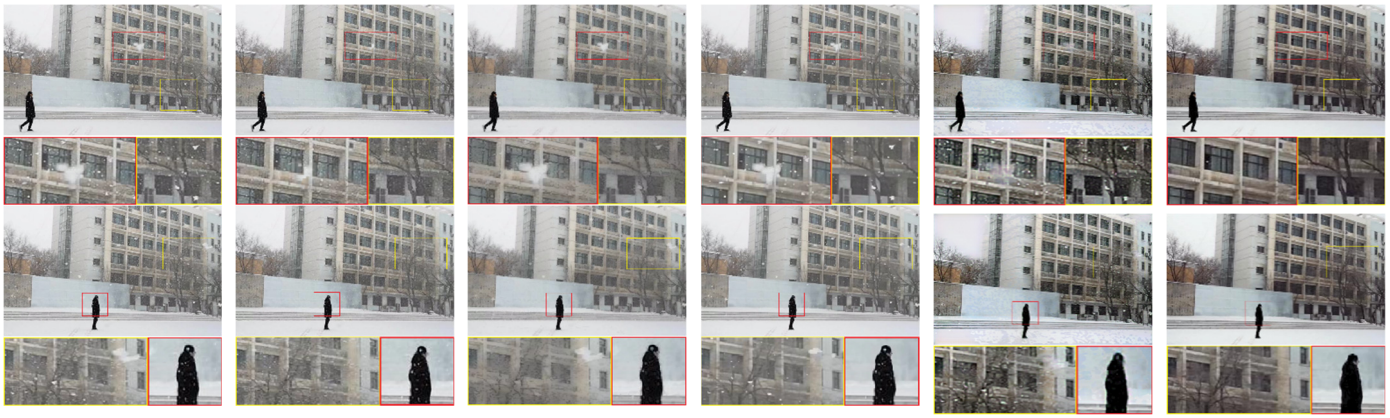
Figure 8 is taken from the snow scene with a pedestrian passing by the static camera, and the snow in front of the black clothes is very obvious. None of the four comparison algorithms remove the snow in front of the background. The method of Kim et al. [16] removes almost all the snow in the background, but unfortunately, the snow in front of the moving object is not removed. The method of Chen et al. [32] can effectively remove snow in front of the moving object, but this method causes distortion of the ground and sky. Our method can truly restore the background and moving objects.

(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 6. Comparison on a real snow video. (**a**) Input, (**b**) Kim et al. [16], (**c**) Wang et al. [8], (**d**) Li et al. [14], (**e**) Chen et al. [32], (**f**) proposed method.

(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 7. Comparison on a real rain video. (**a**) Input, (**b**) Kim et al. [16], (**c**) Wang et al. [8], (**d**) Li et al. [14], (**e**) Chen et al. [32], (**f**) proposed method.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

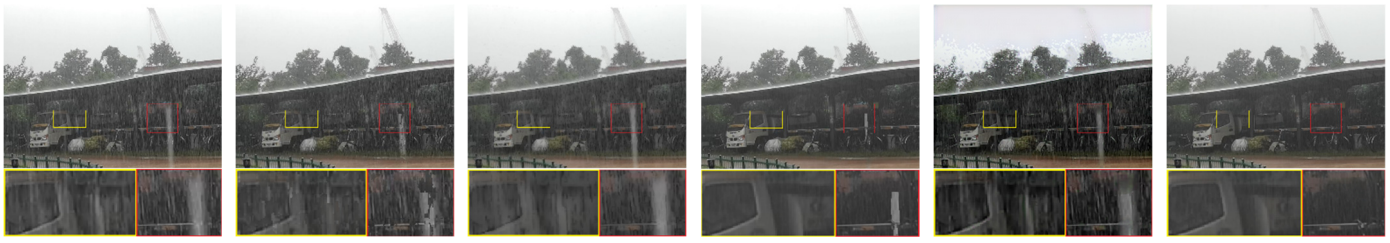
Figure 8. Comparison on a real snow video. (**a**) Input, (**b**) Kim et al. [16], (**c**) Wang et al. [8], (**d**) Li et al. [14], (**e**) Chen et al. [32], (**f**) proposed method.

Figure 9 shows a rainfall scene. The dark background highlights the bright rain streaks, which makes it more difficult to remove the rain. Because the rain streaks are very dense, the matrix completion of Kim et al. [16] cannot remove the dense rain streaks. The method of Li et al. [14] removes the dense rain streaks but does not completely remove the sparse rain streaks. The method of Wang et al. [8] limits the sparse rain streaks removability. Our proposed method removes almost all dense and sparse rain streaks.

Figure 10 is taken from the snow video containing swinging branches and a girl wearing a breathing mask. The slight swing of branches and the deformation of pedestrians pose a great challenge to snow removal. The method of Kim et al. [16] only uses the correlation between five frames to remove snow; the lack of the ability to identify moving objects leads to the wrong removal of the white bag, and the lack of an effective graph cut algorithm blurs the pedestrians. The methods of Wang et al. [8], Li et al. [14] and Chen et al. [32] cannot remove all of the snow in front of the pedestrian. In this scene, our results are still the best of the five methods.

Figure 11 is taken from a surveillance video. There are still some rain streaks left in the results of Kim et al. [16] and Li et al. [14]. The method of Wang et al. [8] seriously blurs the background because of its inherent limitations. Our method removes almost all of the rain streaks.

The sparse rain streaks in Figure 12 pose a great challenge to the desnowing and deraining algorithms. The methods of Wang et al. [8], Li et al. [14] and Chen et al. [32] cannot remove the sparse rain streaks. The method of Kim et al. [16] limits its capability to address this continuous rain streaks. Comparatively, our proposed method still attains promising visual effect in rain removal.

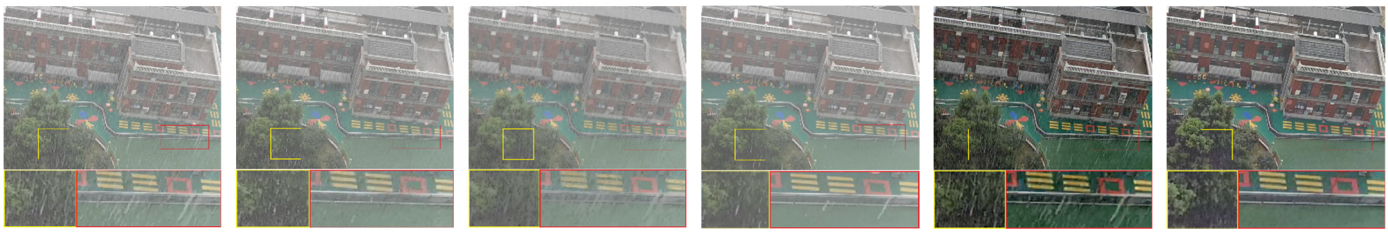


(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 9. Comparison on a real rain video. (**a**) Input, (**b**) Kim et al. [16], (**c**) Wang et al. [8], (**d**) Li et al. [14], (**e**) Chen et al. [32], (**f**) proposed method.

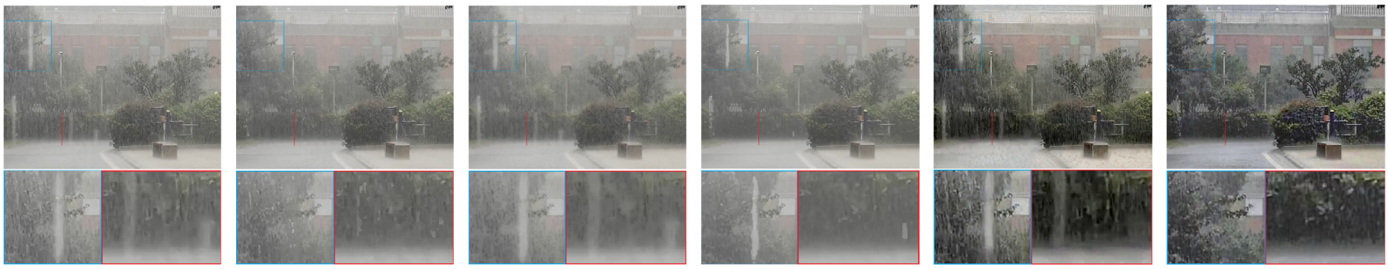
(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 10. Comparison on a real snow video. (**a**) Input, (**b**) Kim et al. [16], (**c**) Wang et al. [8], (**d**) Li et al. [14], (**e**) Chen et al. [32], (**f**) proposed method.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 11. Comparison on a real rain video. (**a**) Input, (**b**) Kim et al. [16], (**c**) Wang et al. [8], (**d**) Li et al. [14], (**e**) Chen et al. [32], (**f**) proposed method.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 12. Comparison on a real rain video. (**a**) Input, (**b**) Kim et al. [16], (**c**) Wang et al. [8], (**d**) Li et al. [14], (**e**) Chen et al. [32], (**f**) proposed method.

4.3. Time Complexity Analysis

In this section, we discuss the runtime of our proposed algorithm and two video desnowing and deraining algorithms [14,16] for dealing with the synthetic snow video (Figure 4) and the real rain video (Figure 11). The resolution of the synthetic snow video is 360×240, and the resolution of the real rain video is 640×480. The number of frames in both videos is 100.

As can be seen from Figure 13, the method of Kim et al. [16] takes the longest time, mainly because it needs to calculate the snow or rain mask maps of each frame before removing the snow or rain. It takes about 50% to 70% of the whole time to calculate the snow or rain mask maps. The runtime of the method of Li et al. [14] is only lower than that of Kim et al. [16]; one of the main reasons is that it needs to learn parameters online.

Whether the processing object is the real rain video or the synthetic snow video, our method is the most efficient.

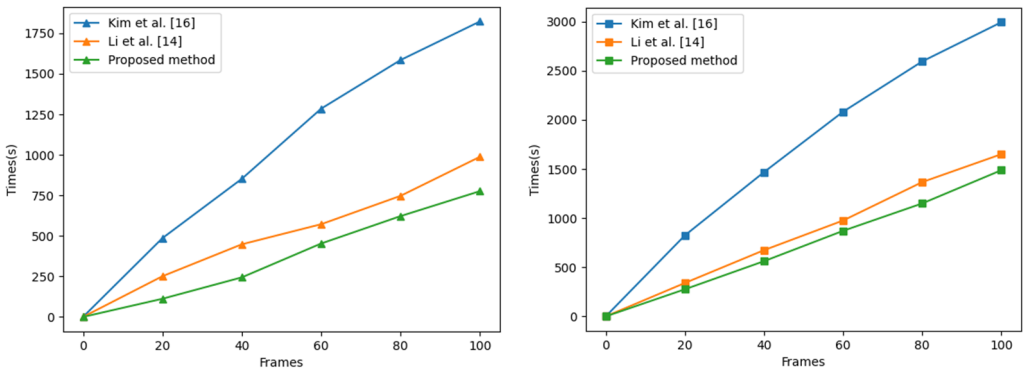


Figure 13. Runtime comparison of comparable methods on two videos. (**a**) The test object is the synthetic snow video (Figure 4). (**b**) The test object is the real rain video (Figure 11).

5. Discussion

In contrast to the tensor decomposition in the literature [25], where the direction property of rain streaks is considered, because snowflakes do not have directional properties, our decomposition method uniformly regards sparse and dense snow and rain as sparse components when decomposing tensors. It can remove snowflakes and rain streaks at the same time since the snowflakes and rain streaks are always intrinsically sparser than the static and quasi-static backgrounds.

From the comparative experiments, it can be seen that the method of Wang et al. [8] is not suitable for snow samples with rich texture in the background. Regardless of how delicate the filtering is, the texture details of the background will be lost, and the dual adaptive spatiotemporal filtering proposed by us is no exception. The failure of Wang et al. [8] lies in global filtering. Unlike their method, our filtering works locally. Generally, the moving object occupies a very small part of the image, and the structure of the moving object is singular, which greatly avoids the loss of texture information caused by filtering.

We tested the performance of the method in over 30 different complex snowfall and rainfall scenes, including different light intensities and different intensities of snowfall and rainfall. The overall performance is good, but there are still three limitations. First, our algorithm works very well with videos taken by stationary or slow-moving cameras (such as surveillance), but it cannot address videos taken by fast moving cameras, due to the lack of video frame alignment technology. Second, when there are other saliency objects in the video background, the quality of the saliency image is reduced, and then the accuracy of moving object detection is affected. Furthermore, when the photometric similarity of moving objects and snowflakes is too high, snowflakes tend to be detected as moving objects. Third, if the moving object's speed is too high, the SIFT matching may only match three to four frames, and the effect of the time domain minimum filtering is not good. In addition, too small moving objects may also lead to the failure of moving object detection. We will further endeavor on these degenerated cases for video snow and rain removal in our future research.

6. Conclusions

With the existing video snow and rain removal methods, it is difficult to meet the demands of outdoor vision sensing systems; one of the main reasons is that, using them, it is difficult to distinguish between sparse snowflakes/rain streaks and moving objects. To solve this problem, in this paper, we utilize tensor decomposition to remove sparse and dense snowflakes and rain streaks from the background, which makes good use of

the spatial location information and the correlation between the three channels of the color video. Moving objects without rich texture information are easily confused with sparse snowflakes. By introducing salience information, the ability of moving object detection is improved. We use feature point matching to obtain the redundant information of the moving object between continuous frames, and then remove snow and rain in front of the moving object by the dual adaptive minimum filtering in the spatiotemporal domain. The experimental results show that our proposed method is superior to other state-of-the-art snow and rain removal methods.

In future research, we will seek more subtle saliency maps to further improve the ability to detect moving objects in snow and rain videos. The existing video snow and rain removal methods cannot effectively address snowfall and rainfall scenes with dynamic backgrounds. We will try to introduce video frame alignment technology [50] to address the snow and rain videos captured by mobile cameras.

Author Contributions: Conceptualization, Y.L.; methodology, Y.L.; software, Y.L.; validation, Y.L., R.W. and Z.J.; formal analysis, Z.J.; writing—original draft preparation, Y.L.; writing—review and editing, Y.L. and Z.J.; supervision, Z.J., J.Y. and N.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science Foundation of China (No. U1803261) and the International Science and Technology Cooperation Project of the Ministry of Education of the People's Republic of China (No. 2016–2196).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: We have evaluated our proposed method on publicly available datasets: the changedetection.net (CDnet) dataset. http://www.changedetection.net (accessed on 29 October 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Garg, K.; Nayar, S.K. Detection and removal of rain from videos. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), Washington, DC, USA, 27 June–2 July 2004; pp. I-I. doi:10.1109/CVPR.2004.1315077
2. Zhang, X.; Li, H.; Qi, Y.; Leow, W.K.; Ng, T.K. Rain removal in video by combining temporal and chromatic properties. In Proceedings of the 2006 IEEE International Conference on Multimedia and Expo, Toronto, ON, Canada, 9–12 July 2006; pp. 461–464.
3. Brewer, N.; Liu, N. Using the shape characteristics of rain to identify and remove rain from video. In Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), Orlando, FL, USA, 4–6 December 2008; pp. 451–458.
4. Park, W.-J.; Lee, K.-H. Rain removal using Kalman filter in video. In Proceedings of the 2008 International Conference on Smart Manufacturing Application, Goyangi, Korea, 9–11 April 2008; pp. 494–497.
5. Pei, S.-C.; Tsai, Y.-T.; Lee, C.-Y. Removing rain and snow in a single image using saturation and visibility features. In Proceedings of the 2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), Chengdu, China, 14–18 July 2014; pp. 1–6.
6. Ding, X.; Chen, L.; Zheng, X.; Huang, Y.; Zeng, D. Single image rain and snow removal via guided L0 smoothing filter. *Multimedia Tools Appl.* **2016**, *75*, 2697–2712.
7. Xu, J.; Zhao, W.; Liu, P.; Tang, X. Removing rain and snow in a single image using guided filter. In Proceedings of the 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE), Zhangjiajie, China, 25–27 May 2012; pp. 304–307.
8. Wang, Y.; Liu, S.; Chen, C.; Zeng, B. A hierarchical approach for rain or snow removing in a single color image. *IEEE Trans. Image Process.* **2017**, *26*, 3936–3950.
9. Shen, Y.; Ma, L.; Liu, H.; Bao, Y.; Chen, Z. Detecting and extracting natural snow from videos. *Inf. Process. Lett.* **2010**, *110*, 1124–1130.
10. Huiying, D.; Xuejing, Z. Detection and removal of rain and snow from videos based on frame difference method. In Proceedings of the 27th Chinese Control and Decision Conference (2015 CCDC), Qingdao, China, 23–25 May 2015; pp. 5139–5143.

11. Yang, T.; Nsabimana, V.; Wang, B.; Sun, Y.; Cheng, X.; Dong, H.; Qin, Y.; Zhang, B.; Ingrabire, F. Snow fluff detection and removal from video images. In Proceedings of the IECON 2017—43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, China, 29 October–1 November 2017; pp. 3840–3844.
12. Wei, W.; Yi, L.; Xie, Q.; Zhao, Q.; Meng, D.; Xu, Z. Should we encode rain streaks in video as deterministic or stochastic? In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2516–2525.
13. Ren, W.; Tian, J.; Han, Z.; Chan, A.; Tang, Y. Video desnowing and deraining based on matrix decomposition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4210–4219.
14. Li, M.; Cao, X.; Zhao, Q.; Zhang, L.; Meng, D. Online rain/snow removal from surveillance videos. *IEEE Trans. Image Process.* **2021**, *30*, 2029–2044.
15. Tian, J.; Han, Z.; Ren, W.; Chen, X.; Tang, Y. Snowflake removal for videos via global and local low-rank decomposition. *IEEE Trans. Multimedia* **2018**, *20*, 2659–2669.
16. Kim, J.-H.; Sim, J.-Y.; Kim, C.-S. Video deraining and desnowing using temporal correlation and low-rank matrix completion. *IEEE Trans. Image Process.* **2015**, *24*, 2658–2670.
17. Yang, Y.; Loquercio, A.; Scaramuzza, D.; Soatto, S. Unsupervised moving object detection via contextual information separation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 879–888.
18. Javed, S.; Mahmood, A.; Al-Maadeed, S.; Bouwmans, T.; Jung, S.K. Moving object detection in complex scene using spatiotemporal structured-sparse RPCA. *IEEE Trans. Image Process.* **2018**, *28*, 1007–1022.
19. Pang, Y.; Ye, L.; Li, X.; Pan, J. Incremental learning with saliency map for moving object detection. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *28*, 640–651.
20. Hu, W.; Yang, Y.; Zhang, W.; Xie, Y. Moving object detection using tensor-based low-rank and saliently fused-sparse decomposition. *IEEE Trans Image Process.* **2016**, *26*, 724–737.
21. Xu, M.; Liu, B.; Fu, P.; Li, J.; Hu, Y.H.; Feng, S. Video salient object detection via robust seeds extraction and multi-graphs manifold propagation. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 2191–2206.
22. Barnum, P.; Kanade, T.; Narasimhan, S. Spatio-temporal frequency analysis for removing rain and snow from videos. In Proceedings of the First International Workshop on Photometric Analysis for Computer Vision—PACV 2007, Rio de Janeiro, Brazil, 14–21 October 2007; 8 p.. https://hal.inria.fr/PACV2007/inria-00264716v1 (accessed on 14 November 2021)
23. Bossu, J.; Hautiere, N.; Tarel, J.-P. Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *Int. J. Comput. Vis.* **2011**, *93*, 348–367.
24. Islam, M.R.; Paul, M. Video Rain-Streaks Removal by Combining Data-Driven and Feature-Based Models. *Sensors* **2021**, *21*, 6856.
25. Jiang, T.-X.; Huang, T.-Z.; Zhao, X.-L.; Deng, L.-J.; Wang, Y. A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4057–4066.
26. Jiang, T.-X.; Huang, T.-Z.; Zhao, X.-L.; Deng, L.-J.; Wang, Y. Fastderain: A novel video rain streak removal method using directional gradient priors. *IEEE Trans. Image Process.* **2018**, *28*, 2089–2102.
27. Li, M.; Xie, Q.; Zhao, Q.; Wei, W.; Gu, S.; Tao, J.; Meng, D. Video rain streak removal by multiscale convolutional sparse coding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6644–6653.
28. Yi, L.; Zhao, Q.; Wei, W.; Xu, Z. Robust online rain removal for surveillance videos with dynamic rains. *Knowl.-Based Syst.* **2021**, *222*, 107006.
29. Zheng, X.; Liao, Y.; Guo, W.; Fu, X.; Ding, X. Single-image-based rain and snow removal using multi-guided filter. In Proceedings of the International Conference on Neural Information Processing, Daegu, Korea, 3–7 November 2013; pp. 258–265.
30. Qian, R.; Tan, R.T.; Yang, W.; Su, J.; Liu, J. Attentive generative adversarial network for raindrop removal from a single image. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2482–2491.
31. Ren, Y.; Nie, M.; Li, S.; Li, C. Single Image De-Raining via Improved Generative Adversarial Nets. *Sensors* **2020**, *20*, 1591.
32. Chen, W.-T.; Fang, H.-Y.; Ding, J.-J.; Tsai, C.-C.; Kuo, S.-Y. JSTASR: Joint size and transparency-aware snow removal algorithm based on modified partial convolution and veiling effect removal. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 754–770.
33. Jaw, D.-W.; Huang, S.-C.; Kuo, S.-Y. DesnowGAN: An efficient single image snow removal framework using cross-resolution lateral connection and GANs. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *31*, 1342–1350.
34. Liu, Y.-F.; Jaw, D.-W.; Huang, S.-C.; Hwang, J.-N. DesnowNet: Context-aware deep network for snow removal. *IEEE Trans. Image Process.* **2018**, *27*, 3064–3073.
35. Li, P.; Yun, M.; Tian, J.; Tang, Y.; Wang, G.; Wu, C. Stacked dense networks for single-image snow removal. *Neurocomputing* **2019**, *367*, 152–163.
36. Liu, Y.; Long, Z.; Huang, H.; Zhu, C. Low CP rank and tucker rank tensor completion for estimating missing components in image data. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 944–954.
37. Chen, Y.; Xiao, X.; Peng, C.; Lu, G.; Zhou, Y. Low-rank tensor graph learning for multi-view subspace clustering. *IEEE Trans. Circuits Syst. Video Technol.* **2021**.

38. Wang, W.; Aggarwal, V.; Aeron, S. Tensor train neighborhood preserving embedding. *IEEE Trans. Signal Process.* **2018**, *66*, 2724–2732.
39. Chen, Y.; Huang, T.-Z.; He, W.; Yokoya, N.; Zhao, X.-L. Hyperspectral image compressive sensing reconstruction using subspace-based nonlocal tensor ring decomposition. *IEEE Trans. Image Process.* **2020**, *29*, 6813–6828.
40. Lu, C.; Feng, J.; Chen, Y.; Liu, W.; Lin, Z.; Yan, S. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 925–938.
41. Lu, C.; Feng, J.; Chen, Y.; Liu, W.; Lin, Z.; Yan, S. Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5249–5257.
42. Xu, H.; Zheng, J.; Yao, X.; Feng, Y.; Chen, S. Fast Tensor Nuclear Norm for Structured Low-Rank Visual Inpainting. *IEEE Trans. Circuits Syst. Video Technol.* **2021**.
43. Cai, Y.; Dai, L.; Wang, H.; Chen, L.; Li, Y. A novel saliency detection algorithm based on adversarial learning model. *IEEE Trans. Image Process.* **2020**, *29*, 4489–4504.
44. Ji, W.; Li, X.; Wei, L.; Wu, F.; Zhuang, Y. Context-aware graph label propagation network for saliency detection. *IEEE Trans. Image Process.* **2020**, *29*, 8177–8186.
45. Zha, Z.-J.; Wang, C.; Liu, D.; Xie, H.; Zhang, Y. Robust deep co-saliency detection with group semantic and pyramid attention. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 2398–2408.
46. Goyette, N.; Jodoin, P.-M.; Porikli, F.; Konrad, J.; Ishwar, P. Changedetection. net: A new change detection benchmark dataset. In Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA, 16–21 June 2012; pp. 1–8.
47. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612.
48. Zhang, L.; Zhang, L.; Mou, X.; Zhang, D. FSIM: A feature similarity index for image quality assessment. *IEEE Trans. Image Process.* **2011**, *20*, 2378–2386.
49. Sheikh, H.R.; Bovik, A.C. Image information and visual quality. *IEEE Trans. Image Process.* **2006**, *15*, 430–444.
50. Zeng, X.; Howe, G.; Xu, M. End-to-End Robust Joint Unsupervised Image Alignment and Clustering. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 3854–3866.