Robust Video Content Alignment and Compensation for Clear Vision Through the Rain

Jie Chen\textsuperscript{\$}, Member, IEEE, Cheen-Hau Tan\textsuperscript{\$}, Member, IEEE, Junhui Hou, Member, IEEE, Lap-Pui Chau, Fellow, IEEE, and He Li,

Abstract—Outdoor vision-based systems suffer from atmospheric turbulences, and rain is one of the worst factors for vision degradation. Current rain removal methods show limitations either for complex dynamic scenes, or under torrential rain with opaque occlusions. We propose a novel derain framework which applies superpixel (SP) segmentation to decompose the scene into depth consistent units. Alignment of scene contents are done at the SP level, which proves to be robust against rain occlusion interferences and fast camera motion. Two alignment output tensors, i.e., optimal temporal match tensor and sorted spatial-temporal match tensor, provide informative clues for the location of rain streaks and the occluded background contents. Different classical and novel methods such as Robust Principle Component Analysis and Convolutional Neural Networks are applied and compared for their respective advantages in efficiently exploiting the rich spatial-temporal features provided by the two tensors. Extensive evaluations show that advantage of up to 5dB is achieved on the scene restoration PSNR over state-of-the-art methods, and the advantage is especially obvious with highly complex and dynamic scenes. Visual evaluations show that the proposed framework is not only able to suppress heavy and opaque occluding rain streaks, but also large semi-transparent regional fluctuations and distortions.

Index Terms—superpixel segmentation, rain removal, dynamic content, spatial-temporal filtering, robust principle component analysis, convolutional neural networks.

1 INTRODUCTION

As modern intelligent systems rely more and more on visual information, the robustness of the system becomes dependent on the quality of visual input. Outdoor vision modules, in particular, are susceptible to visual degradations caused by atmospheric turbulences. Turbulent weather conditions can be generally classified into steady or dynamic systems \cite{1, 2}. For steady systems, such as fog and haze, small (1-10 $\mu$m) atmospheric particles suspended in the air induce accumulated scattering of scene radiance; hence, they primarily degrade visual contrast and visibility. For dynamic systems, such as rain and snow, the atmospheric particles are much larger (0.1-10 mm) and relatively fast-moving. As such, the motions of individual or aggregate particles become visible and cause complex visual distortions. These distortions could unintentionally introduce motion content or obscure visual features, which negatively impacts computer vision algorithms. In this paper, we propose a rain removal framework that handles both the dynamic and steady state visual degradations of rain.

To devise a rain removal algorithm, we should first understand the visual effects caused by rain. Rain appearances can be significantly different depending on scene distance ranges and imaging configurations \cite{3}. Based on their visual effects, we classify rain into three types. As shown in Fig. 1, these are:

- **Occluding rain:** these raindrops fall within the depth of field of a camera. They are in focus, and fall at high speed. Hence, they appear as slender streaks under normal camera exposure settings (shutter speed slower than 1/200 second). In general, these streaks are somewhat opaque and occlude the background scene.

- **Veiling rain:** raindrops that fall nearer than the current depth of field cover a larger field of view and are out of focus. They might not take distinct shapes and resemble semi-transparent veils, causing regional intensity fluctuations and distortions.

- **Accumulated rain:** raindrops distributed further than the current depth of field are out of focus, cover a tiny field of view, and are distributed over large distances. Their visual effects are accumulated over many samples and are stable over time, producing visual degradations similar to that of fog and haze.

Based on the types of input, we can categorize rain removal methods into image-based methods and video-based methods. Image-based methods rely only on information from the processed frame while video-based methods have access to other frames in the sequence. Most video-based methods rely primarily on temporal information since they are very discriminative for rain detection — both occluding rain and veiling rain produce intensity fluctuations over time; other clues like rain structure might not consistently hold under all rainy conditions, e.g., under veiling rain.
Fig. 1: Different photometric appearances of raindrops at different distance range and camera configurations.

However, temporal information are reliable only if, for each pixel, the corresponding scene content is consistent over time. This constraint could be violated due to two factors – motion of the camera and presence of moving objects. Previous works tackle these two issues separately. Camera motion-induced scene content shifts were reversed using global frame alignment [4], [5], [6]. However, the granularity of global alignment is too large when scene depth range is large; to be aligned accurately, scene content of different depths should be independently aligned. Thus, by using global alignment, parts of the scene could be poorly aligned.

The presence of moving objects disrupts temporal consistency of scene content, and could thus cause scene elements to be misclassified as rain. Earlier works attempt to identify the misclassified scene pixels by testing various properties of candidate rain pixels, such as photometric properties [7], chroma changes [8], geometric properties [9] or local statistics [4]. This approach, however, does not adequately remove rain that overlap moving objects. Later works first segment out motion regions or foreground objects before applying rain removal on these regions separately [6], [10], [11]. However, these methods rely on accurate segmentation of foreground regions.

We recognize that both camera motion and moving objects cause scene content to be misaligned and thus corrupt temporal information. Hence, in this paper, we propose a novel and elegant framework that simultaneously solves both issues by aligning scene content to compensate for their misalignment – superpixel-based content alignment and compensation (SPAC) for rain removal. First, the target video frame is segmented into superpixels (SPs); each SP corresponds to a depth consistent patch of scene content. Then, for each target SP, we form tensors by stacking patches similar to the target SP. This step aligns content correspond-

ing to both scene background and moving objects without prior segmentation of the moving object. Scene content is also much better aligned at a SP-level granularity. Finally, a de-rained SP patch is extracted from the tensors. Within this framework, we experimented on several ways to utilize these tensors for rain removal – taking the average values of the tensor (SPAC-Avg and SPAC-\(F_1\)) , applying robust principle analysis (SPAC-RPCA), and processing it using a Convolutional Neural Network (SPAC-CNN).

Extensive evaluations show that up to 5dB reconstruction PSNR advantage is achieved over state-of-the-art methods, and the advantage is especially obvious in highly dynamic scenes taken from a fast moving camera. Visual evaluations show that the proposed framework is not only able to remove heavy and opaque occluding rain streaks, but also suppress large semi-transparent regional fluctuations and distortions. Fig. 2 illustrates the advantage of our proposed algorithm over existing methods in a challenging video sequence. The contributions of this work can be generalized as follows.

- We propose to apply spatial-temporal alignment on scene content for video-based rain removal. This approach, ensures content consistency over consecutive frames whether subjected to camera motion or presence of moving objects. Hence, segmentation of moving objects for separate treatment is not required.
- We propose to use Superpixels as the basic unit for scene content alignment. Alignment of small blocks of pixels is not robust to structured noise, especially large rain streaks. Large blocks of pixels, however, are oblivious to object boundaries, and might encompass disjointed content located at different depths. By using SPs, we can form sufficiently large clusters of pixels that also respect object boundaries.
- We tested a set of different rain removal methods on aligned scene content. In particular, we propose a neural network-based (SPAC-CNN) method where a set of efficient spatial-temporal features is used to help restore high frequency details lost during the de-raining process.
- As far as we know, this is the first work to categorize testing rain data based on camera speed for separate evaluations. A rain video dataset was created with a car-mount camera with speed range between 0 to 30 km/h over different dynamic scenes. The data set was used both for model training and evaluations.

A shorter version of this work appeared in [15]. This journal version presents more explanations and technical details, and includes additional experiments to verify the efficiency of using superpixels over blocks. We have also included additional rain removal methods, SPAC-\(F_1\) and SPAC-RPCA, under our superpixel-based content alignment framework. Comparisons to additional competing video-based methods have also been added. A patent for an implementation based on SPAC-\(F_1\) [16] is being filed in Singapore.

The remainder of the paper is organized as follows. In Section 2 we will discuss the current rain removal methods. Then, in Section 3 we introduce our proposed framework and methods in detail. In Section 4 we show comprehensive
Fig. 2: Comparison of derain outputs by different algorithms for a challenging video sequence with fast camera motion and heavy rain fall. Image-based derain methods, i.e., discriminative sparse coding (DSC) [12] and deep detail network (DDN) [13], fail to remove large and/or opaque rain streaks. Video-based derain methods, i.e., via matrix decomposition (VMD) [6], and via temporal correlation and low-rank completion (TCL) [14], create serious blur due to fast camera motion, and also show significant amount of remnant rain streaks. Our proposed SPAC-CNN can cleanly remove the rain streaks and preserve scene content truthfully.

Chen and Hsu [26] derived an optimization function where that the rain layer is assumed to be low ranked while the background layer varies smoothly. Luo et al. [12] relied on discriminative sparse coding, where their rain dictionary is trained from simulated rain. Li et al. [27], [28] modeled rain and background layers using Gaussian mixture models (GMM). Natural images are used to train the background GMM while pure rain streak regions in the input, which do not contain background features, are used for training the rain GMM. Instead of applying layer separation, Kim et al. [29] identified rain pixels using local statistical and geometric properties, such as orientation, shape and brightness, and then recovered their values using non-local means filtering.

Convolutional Neural Networks have been very effective in both high-level vision tasks [30] and low-level vision applications for capturing signal characteristics [31], [32]. Different network structures and features were explored for rain removal. Fu et al. proposed the deep detail network [13] which uses a priori image domain knowledge to focus on learning high frequency details of the rain and differentiating them from interferences of background structures. Yang et al. [33] proposed a multi-task deep learning architecture that involves learning of the binary rain streak map, the appearance of rain streaks, and the clean background under one framework. These CNN structures still shown limitations for opaque and heavy rain occlusions.

2.2 Video-based methods

Most video-based methods rely primarily on temporal analysis of pixel values. In principle, the intensity of each pixel is analyzed over time; elevated intensity values indicate the presence of rain. These rain pixels are then restored using pixel values from neighboring frames. The issue with temporal analysis is that camera motion and dynamic scene motion could also cause intensity spikes, creating false pos-
two immediate neighbors using optical flow. This frame, difficult to achieve under fast camera motion. Furthermore, a relatively large number of frames is required for low rank decomposition, which makes their alignment constrained to be sparse. Moving objects are de-rained by temporally aligning them using patch matching, while are constrained to be low-ranked, and foreground and rain layers are assumed to be rain-free, is compared with the target frame to detect rain candidates, which are then refined using sparse coding. Rain pixels are recovered using block-based matrix completion. There are also works with less emphasis on using temporal data. Barnum et al. [35] modeled rain streaks in frequency space, Santhaseelan and Asari [36] use phase congruency on frame difference images to detect candidate rain streaks, and Kim et al. [37] applied horizontal directional thresholding to detect candidate rain streaks.

In general, video-based methods potentially perform better than image-based methods, since temporal information is available. However, unless accounted for, camera motion and scene object motion invalidate temporal information and thus affect performance. Image-based methods, as do some video-based methods, rely on priors or assumptions about properties of rain. However, as shown in our real world experimental videos, rain could appear in irregular sizes, shapes, brightness, opacity, or distributions; these rain are not easily removed by existing image-based methods.

3 Proposed Model

Throughout the paper, scalars are denoted by italic lowercase letters, 2D matrices by upper-case letters, and 3D tensors and functions by script letters.

Fig. 3 shows the system diagram for the proposed rain removal framework based on SuperPixel content Alignment and Compensation (SPAC). The framework can be divided into two parts: first, video content alignment is carried out based on SP units. For each SP, we perform two SP template matching operations, which produce two output tensors: the optimal temporal match tensor $T_0$ and the sorted spatial-temporal match tensor $T_1$. Second, rain removal and scene content compensation are carried out over the slices of the two tensors $T_0$ and $T_1$. Various classical and novel methods such as Robust Principle Component Analysis (RPCA) and Convolutional Neural Networks (CNN), will be investigated and compared under our proposed SPAC framework [15] for their respective advantages in the task of scene content restoration.

1. A slice is a two-dimensional section of a higher dimensional tensor, defined by fixing all but two indices. 

---

**Fig. 3:** System diagram for the proposed SPAC framework. The switch $S_1$ changes algorithms between SPAC-Arg, SPAC-$F_1$, SPAC-RPCA, and SPAC-CNN.
3.1 Robust Content Alignment via Superpixel Spatial-Temporal Matching

Given a target frame $I_0$ of a rainy video, we consider its immediate past and future neighbor frames to create a sliding buffer window of length $n_t$: $[I_i | i = [-\frac{n_t-1}{2}, \frac{n_t-1}{2}]].$ Here, negative and positive $i$ indicate past and future frames, respectively. We only derain the Y luminance channel. The derain output is used to update the corresponding frame in the buffer to improve rain removal of future target frames (As shown Fig. 3). This history buffer update mechanism contributes to cleaner derain output especially under heavy rainfall scenarios.

One of the most important procedures for video-based derain algorithms are the estimation of content correspondence between video frames. With accurate content alignment, rain occlusions could be easily detected and removed with information from the temporal axis. The alignment, however, is no trivial task. Camera motion changes the scene significantly, and the change of perspective causes scene contents at different distance to shift at different parallax. Additionally, scene contents could have their own rigid/non-rigid motion, not to mention the serious interference caused by rain streak occlusions. All these factors make content alignment extremely challenging for rainy videos.

3.1.1 Content Alignment: Global vs. Superpixel

The popular solution to compensate camera motion between two frames is via a homography transform matrix estimated based on global consensus of a group of matched feature points [40], [41]. Due to the reasons analyzed in Sec. 1, perfect content alignment can rarely be achieved for all pixels with a global transform at whole frame level, especially for dynamic scenes with large depth range. To illustrate, two frames in Fig. 4(a) and (b) are geometrically aligned via global homography transform, and the aligned frames are overlapped and shown in Fig. 4(c). It can be seen that scene contents at far and middle distance ranges are well-aligned; however, for nearer contents (e.g., the banisters), obvious mis-alignment appear.

The solution naturally turns to pixel-level alignment, which faces no fewer challenges: first, feature points are sparse, and feature-less regions are difficult to align; more importantly, rain streak occlusions will cause serious interferences to feature matching at single pixel level. Contextual information from neighboring area is indispensable to overcome interference. This leads us to our solution: to decompose images into smaller depth consistent units.

The concept of SuperPixel (SP) is to group pixels into perceptually meaningful atomic regions [42], [43], [44]. Boundaries of SP usually coincide with those of the scene contents. Comparing Fig. 4(d) and (e), the SPs are very adaptive in shape, and are more likely to segment uniform depth regions compared with rectangular units. We adopt SP as the basic unit for content alignment.

3.1.2 Optimal Temporal Matching for Rain Detection

Let $P_k$ denote the set of pixels that belong to the $k$-th SP on $I_0$. Let $X_k \in \mathbb{R}^{n_x \times n_y}$ be the bounding box that covers all pixels in $P_k$ ($P_k \subset X_k$). Let $B_k \in \mathbb{R}^{n_s \times n_x \times n_y}$ denote a spatial-temporal buffer centered on $P_k$. As illustrated in Fig. 5, the temporal range of $B_k$ spans the entire sliding buffer window, and its spatial range $n_s \times n_x$ is set to cover the possible motion range of $P_k$ in its neighboring frames.

Pixels within the same SP are very likely to belong to the same object and show identical motion between adjacent frames. Therefore, we can approximate the SP’s appearance in other frames based on its appearance in the current frame via linear translations.

Searching for the reference SP is done by template matching of the target SP at all candidate locations in $B_k$. A match location is found at frame $I_t$ according to:

$$
(\hat{u}, \hat{v}) = \arg \min_{u,v} \sum_{(x,y) \in X_k} |B_k(x + u, y + v, t') - X_k(x, y)|^2 \odot M_{SP}(x, y).
$$

As shown in Fig. 5(d), $M_{SP}$ indicates SP pixels $P_k$ in the bounding box $X_k$. $\odot$ denotes element-wise multiplication. Each match at a different frame becomes a temporal slice for the optimal temporal match tensor $T_0 \in \mathbb{R}^{n_x \times n_y \times n_t}$:

$$
T_0(\cdot, \cdot, t') = B_k(x + \hat{u}, y + \hat{v}, t'), \ (x, y) \in X_k.
$$

Based on the temporal clues provided by $T_0$, a rain mask can be estimated. Since rain increases the intensity of its covered pixels [7], rain pixels in $X_k$ are expected to have higher intensity than their colocated temporal neighbors in $T_0$. We first compute a binary tensor $M_0 \in \mathbb{R}^{n_x \times n_y \times n_t}$ to detect positive temporal fluctuations:

$$
M_0 = \begin{cases} 
1 & R(X_k, n_t) - T_0 \geq \epsilon_{rain}, \\
0 & \text{otherwise}
\end{cases}
$$

where operator $R(\Phi, \psi)$ is defined as replicating the 2D slices $\Phi \in \mathbb{R}^{n_1 \times n_2}$ $\psi$ times and stacking along the third dimension into a tensor of $\mathbb{R}^{n_1 \times n_2 \times \psi}$. To robustly handle re-occurring rain streaks, we classify pixels as rain when at least 3 positive fluctuations are detected in $M_0$. An initial rain mask $M_{rain} \in \mathbb{R}^{n_x \times n_y}$ can be calculated as:
\[ \hat{M}_{\text{rain}}(x, y) = \sum_{t} M_t(x, y, t) \geq 3. \tag{4} \]

Due to possible content mis-alignment, edges of background could be misclassified as rain. Since rain streaks don’t affect values in the chroma channels (Cb and Cr), a rain-free edge map \( M_e \) could be calculated by thresholding the sum of gradients of these two channels with \( \epsilon_e \). The final rain mask \( \hat{M}_{\text{rain}} \in \mathbb{R}^{n_x \times n_z} \) is calculated as:

\[ \hat{M}_{\text{rain}} = \hat{M}_{\text{rain}} \odot (1 - M_e). \tag{5} \]

A conceptual demonstration of \( \hat{M}_{\text{rain}}, M_e, \) and \( M_{\text{rain}} \) is shown in Fig. 5(a), (b), and (c), respectively. In our implementation, \( \epsilon_{\text{rain}} \) is set to 0.012 while \( \epsilon_{\text{edge}} \) is set to 0.2.

3.1.3 Sorted Spatial-Temporal Template Matching for Rain Occlusion Suppression

A second round of template matching will be carried out based on the following cost function:

\[ E(u, v, t) = \sum_{(x, y) \in X_k} |B_k(x + u, y + v, t) - X_k(x, y)|^2 \odot M_{\text{RSP}}(x, y). \tag{6} \]

The rain-free matching template \( M_{\text{RSP}} \) is calculated as:

\[ M_{\text{RSP}} = M_{\text{SP}} \odot (1 - M_{\text{rain}}). \tag{7} \]

As shown in Fig. 5(c), only the rain-free background SP pixels will be used for matching. Candidate locations in \( B_k \) (except current frame \( B_k(\cdot, \cdot, 0) \)) are sorted in ascending order based on their cost \( E \) defined in Eq. (6). The top \( n_{st} \) candidates with smallest \( E \) will be stacked as slices to form the sorted spatial-temporal match tensor \( T_1 \in \mathbb{R}^{n_x \times n_z \times n_{st}} \).

With masked template matching eliminating the interferences from rain, the slices of \( T_1(\cdot, \cdot, t) \) are expected to be well-aligned to the current target SP. Since rain pixels are temporally randomly and sparsely distributed within \( T_1 \), when \( n_{st} \) is sufficiently large, we can get a good estimation of the rain free image through tensor slice averaging, which functions to suppress rain induced intensity fluctuations, and recover the occluded background pixels:

\[ X_{\text{Avg}} = \frac{1}{n_{st}} \sum_{t} T_1(\cdot, \cdot, t). \tag{8} \]

If we only extract one tensor slice from each frame as done for \( T_0 \), there will be too few samples for suppression of heavy rain. For \( T_0 \), we set \( n_{st} > n_1 \); thus, \( T_0 \) contains corresponding scene content from neighboring frames along with their slightly shifted copies. Hence, \( T_0 \) tensor averaging is similar to weighted spatio-temporal averaging of aligned scene content, where the binary weights are dependent on similarity to the target patch. We denote this method as SPAC-Avg, and use it as the baseline method in our SPAC framework. Fig. 6 illustrates the steps involved in computing SPAC-Avg.

3.2 Rain Removal and Low Rank Content Modeling via Robust Principle Component Analysis

As shown in experimental results, although SPAC-Avg proves to be very effective in suppressing rain, it also blurs background details. Hence, in this section, we attempt to use a more advanced mathematical model – Robust Principle Component Analysis (RPCA) [45].

RPCA was originally proposed to recover corrupted entries from data, where the collection of data is low ranked and the corrupted entries are sparse. However, it has been successfully applied to many other applications, such as [46, 47, 48]. Its main advantage over averaging-based methods is that it is robust to corruptions of large magnitudes.

Let \( V(\cdot) : \mathbb{R}^{n_x \times n_z} \rightarrow \mathbb{R}^{n_p \times 1} \) denote the operator that extracts the pixels of a target SP from a \( n_x \times n_z \) 2D rectangular patch, and then stacks them as a vector \( n_p \times 1 \). Here, \( n_p \leq n_x \times n_z \) denotes the number of pixels in the target SP. A 2D matrix \( \Psi \in \mathbb{R}^{n_p \times n_{st}} \) can be formed from the target SP and slices in \( T_1 \):

\[ \Psi = [X_k \mid V(T_1(\cdot, \cdot, 1)) \mid \ldots \mid V(T_1(\cdot, \cdot, n_{st} - 1))]. \tag{9} \]

Should the SP contents in \( T_1 \) be aligned, columns of \( \Psi \) will be similar with each other, and \( \Psi \) should be low-rank. We model the rain-free content of \( \Psi \) as a low-rank matrix \( L \); the rain occlusions and small shifts typically affect only a small fraction of pixels in an image, therefore we model them as a sparse matrix \( R \).

Finally, the rain removal problem can now be expressed as:

\[ \min_{L,R} \text{ rank}(L) + \gamma \| R \|_0, \quad \text{s.t.} \quad \Psi = L + R. \tag{10} \]

Here, the \( \ell_0 \)-norm \( \| \cdot \|_0 \) counts the number of nonzero entries in the matrix \( R \). \( \gamma \) is a parameter that trades off the rank of \( L \) versus the sparsity of \( R \).

The problem in Eq. (10) is highly nonconvex due to the calculation of matrix rank and \( \ell_0 \)-norm. As suggested in [45], we relax it to its convex surrogate as:

\[ \min_{L,R} \| L \|_* + \gamma \| R \|_1, \quad \text{s.t.} \quad \Psi = L + R, \tag{11} \]

which replaces rank with the nuclear norm of \( L \) (which equals the sum of its singular values), and replaces the \( \ell_0 \)-norm \( \| R \|_0 \) with the \( \ell_1 \)-norm: \( \| \cdot \|_1 = \sum_{ij} | \Psi_{i,j} | \). It has been shown in [45] that Eq. (11) can exactly recover \( L \) in Eq. (10) as long
as long as the rank of the matrix $\Psi$ is not too high and the number of nonzero entries in $R$ is not too large. This condition is generally met when the alignment quality of $T_1$ is high, and when the rain occlusions are not too dense.

The convex problem in Eq. (11) can be efficiently solved via modern first-order optimization techniques such as the Augmented Lagrange Multiplier (ALM) algorithm [49]. The derained output $SP_{AC-CNN}$ is extracted from the first column of low-rank matrix $L$ using the inverse of $V(\cdot)$. We denote this method as $SP_{AC-RPCA}$.

### 3.3 Rain Removal and Content Compensation in a CNN framework

As to be shown in Sec. 4 when camera motion gets faster (which leads to larger mis-alignment errors), and when rain occlusions get dense, $SP_{AC-RPCA}$ shows limitations. Under both circumstances, the low-rank assumption of $L$, and the sparseness assumption of $R$ are violated.

We further investigate another solution for this challenging scenario with a CNN model, which is expected to have a better capacity to robustly model the rain-free background contents with its deep and highly non-linear layers. In this section, we aim at designing a CNN structure (denoted as $SP_{AC-CNN}$) for restoring the lost high frequency scene content details back to $X_{Avg}$, however without re-introducing the rain streaks (rain streaks are also high frequency signals). We will first introduce some informative features for this task, then we proceed to the network structure design details.

#### 3.3.1 Occluded Background Feature

$X_{Avg}$ from Eq. 8 can be used as one important clue to recover rain occluded pixels. Rain streak pixels indicated by the rain mask $M_{rain}$ are replaced with corresponding pixels from $X_{Avg}$ to form the first feature $F_1 \in \mathbb{R}^{n_x \times n_x \times 1}$:

$$F_1 = X_k \odot (1 - M_{rain}) + X_{Avg} \odot M_{rain}. \quad (12)$$

Feature $F_1$ itself is already a good derain output, which we denote as $SP_{AC-F_1}$. However its quality is greatly limited by the correctness of the rain mask $M_{rain}$. For false positive rain pixels, $X_{Avg}$ will introduce content detail loss; for false negative pixels, rain streaks will be added back from $X_k$. This calls for more informative features for the CNN.

#### 3.3.2 Temporal Consistency Feature

The temporal consistency feature is designed to handle false negative rain pixels in $M_{rain}$, which falsely add rain streaks back to $F_1$. For a correctly classified and recovered pixel (a.k.a. true positive) in Eq. (12), intensity consistency should hold such that for the collocated pixels in the neighboring frames, there are only positive intensity fluctuations caused by rain in those frames. Any obvious negative intensity drop along the temporal axis is a strong indication that such pixel is a false negative pixel.

The temporal slices in $T_0$ establishes optimal temporal correspondence at each frame, which embeds enough

2. False positive rain pixels refer to background pixels falsely classified as rain; false negative rain pixels refer to rain pixels falsely classified as background.

- In order to facilitate the network training, we limit the mapping range between the input features and regression output by removing the low frequency component ($X_{avg}$) from these input features. Pixels in $X_k$ but outside of the SP $P_k$ is masked out with $M_{SP}$:

$$\hat{F}_1 = (F_1 - X_{Avg}) \odot M_{SP}, \quad (14)$$

$$\hat{F}_2 = (F_2 - R(X_{Avg}, n_t - 1)) \odot R(M_{SP}, n_t - 1), \quad (15)$$

$$\hat{F}_3 = (F_3 - R(X_{Avg}, n_{st})) \odot R(M_{SP}, n_{st}). \quad (16)$$

The final input feature set is $\{\hat{F}_1, \hat{F}_2, \hat{F}_3\}$. The feature preparation process for $SP_{AC-CNN}$ is summarized in Fig. 6

#### 3.3.3 High Frequency Detail Feature

The matched slices in $T_1$ are sorted according to their rain-free resemblance to $X_k$, which provide good reference to the content details with supposedly small mis-alignment. We directly use the tensor $T_1$ as the last group of features $\mathcal{F}_3 = T_1 \in \mathbb{R}^{n_x \times n_x \times n_t \times n_{st}}$. This feature will compensate the detail loss introduced by the operations in Eq. (12) for false positive rain pixels.

In order to facilitate the network training, we limit the mapping range between the input features and regression output by removing the low frequency component ($X_{avg}$) from these input features. Pixels in $X_k$ but outside of the SP $P_k$ is masked out with $M_{SP}$:

$$\hat{F}_1 = (F_1 - X_{Avg}) \odot M_{SP}, \quad (14)$$

$$\hat{F}_2 = (F_2 - R(X_{Avg}, n_t - 1)) \odot R(M_{SP}, n_t - 1), \quad (15)$$

$$\hat{F}_3 = (F_3 - R(X_{Avg}, n_{st})) \odot R(M_{SP}, n_{st}). \quad (16)$$

The final input feature set is $\{\hat{F}_1, \hat{F}_2, \hat{F}_3\}$. The feature preparation process for $SP_{AC-CNN}$ is summarized in Fig. 6

#### 3.3.4 CNN Structure and Training Details

The CNN architecture is designed as shown in Fig. 7. The network consists of four convolutional layers with decreasing kernel sizes of 11, 5, 3, and 1. All layers are followed by a rectified linear unit (ReLU). Our experiments show this fully convolutional network is capable of extracting useful information from the input features and efficiently providing reliable predictions of the content detail $X_{detail} \in \mathbb{R}^{n_x \times n_x \times 1}$. The final rain removal output will be:

$$X_{detain} = X_{Avg} + X_{detail}. \quad (15)$$

For the CNN training, we minimize the $L_2$ distance between the derain output and the ground truth scene:

$$E = [\hat{X} - X_{detain}]^2, \quad (16)$$

Here $\hat{X}$ denotes the ground truth clean image. We use stochastic gradient descent (SGD) to minimize the objective function. Mini-batch size is set as 50 for better trade-off between speed and convergence. The Xavier approach [51] is used for network initialization, and the ADAM solver [52] is adapted for system training, with parameter settings $\beta_1 = 0.9$, $\beta_2 = 0.999$, and learning rate $\alpha = 0.0001$. Fig. 7: CNN architecture for compensation of mis-alignment blur. Each convolutional layer is followed by a rectified linear unit (ReLU).
4 Performance Evaluation

In this section, we evaluate the functional components as well as overall derain performance of the methods under the proposed SPAC framework, and compare them with recent state-of-the-art rain removal methods.

Each frame was segmented into around 300 SPs using the SLIC method [42]. For a VGA resolution frame, this results in average bounding box sizes of $n_x = 32$. By considering two previous and two future neighbor frames, the spatial-temporal buffer $B_k$ dimension was $n_x \times n_x \times n_t = 30 \times 30 \times 5$. The length of tensor $T_1$, $n_{st}$, was set to 10.

For SPAC-RPCA, $\lambda$ is set to $1/\max(\sqrt{n_x}, \sqrt{n_{st}})$, which we found to be able to achieve a good balance between rain removal and background detail preservation. For SPAC-CNN, MatConvNet [54] was adopted for model training, which took approximately 54 hours to converge over the training dataset introduced in Sec. 3.3.4. The training and all subsequent experiments were carried out on a desktop with Intel E5-2650 CPU, 56GB RAM, and NVIDIA GeForce GTX 1070 GPU.

All the image and video results reported in this section can be accessed online[3].

4.1 SP Matching Efficiency Evaluation

For all methods under the proposed SPAC framework, the choice of SP as the basic operation unit is key to their performances. When other decomposition units are used instead (e.g., rectangular patches), matching accuracy deteriorates, and very obvious averaging blur will be introduced especially at object boundaries.

We quantitatively evaluate the advantage of content alignment based on SP units compared with rectangular units over the Middlebury Stereo Dataset [50]. The dataset contains 9 image sequences, with each looking at a target scene from 7 different viewing angles. The baselines between the angles are large enough to simulate fast camera motion.

A straightforward reconstruction method is used for this evaluation. The 7 views are treated as 7 consecutive video frames and forms a 7-frame buffer. For a certain target matching unit (either SP or rectangular block) on the central view (i.e., the 4th frame), we find a best match in the spatial-temporal buffer with the smallest matching error defined in

Table 1: Central view reconstruction PSNR (dB) based on six side views for the Middlebury Stereo Dataset [50].

| Data     | block Matching PSNR | SP Matching PSNR | SP Matching Advantage |
|----------|---------------------|------------------|----------------------|
| Art      | 30.68               | 32.18            | +1.51                |
| Books    | 34.74               | 35.83            | +1.09                |
| Computer | 34.69               | 36.40            | +1.71                |
| Dolls    | 33.02               | 34.50            | +1.48                |
| Drumsticks | 29.09             | 30.96            | +1.87                |
| Dwarves  | 35.55               | 36.80            | +1.25                |
| Laundry  | 33.03               | 33.75            | +0.72                |
| Moebius  | 34.05               | 35.35            | +1.30                |
| Reindeer | 35.15               | 35.42            | +0.27                |
| Average  | 33.33               | 34.58            | +1.24                |

3. Testing dataset and results available at: https://github.com/hotndy/SPAC-SupplementaryMaterials
Fig. 9: 8 testing rainy scenes synthesized with Adobe After Effects [33]. First row (Group a) are taken with a panning unstable camera. Second row (Group b) are from a fast moving camera (speed range between 20 to 30 km/h)

Eq. (6). Note that in this experiment, $M_{gsp}$ is set equal to the SP mask $M_{sp}$, since there is no rain in this dataset; and for rectangular patches, the mask $M_{gsp}$ is directly set to all ones. An approximated central view will be reconstructed using their optimal spatial-temporal matches from the six side views (i.e., frame 1,2,3,5,6,7 of the sequence).

The approximation quality measured in PSNR for each data is shown in TABLE 1. As can be seen, matching based on SP units provides an average 1.24 dB advantage over those based on rectangular patches. Fig. 8 shows the visual comparison of reconstructed central views between the two methods. As can be seen, SP provides better approximation especially along occluding boundaries. Reconstruction based on rectangular patches show obvious artifacts at these regions. This experiment validates the advantage of using SP units for content alignment between adjacent frames.

4.2 Rain Removal Quantitative Evaluations

To quantitatively evaluate our proposed algorithms, we used a set of 8 videos (different from the training set), and synthesized rain over these videos with varying parameters with Adobe After Effects. Each video contains around 200 to 300 frames. All subsequent results shown for each video are the average results for all frames.

To test the algorithm performance in handling cameras with different motion, we divided the 8 testing scenes into two groups: Group a consists of scenes taken from a panning and unstable camera with slow movements; Group b from a fast moving car-mount camera (with speed range between 20 to 30 km/h). Thumbnails and the labeling of each testing scene are shown in Fig. 9. We evaluated all the methods under the SPAC framework, i.e., SPAC-F1, SPAC-Avg, SPAC-RPCA, and SPAC-CNN. Five competing state-of-the-art rain removal methods were chosen for comparison. This includes two image-based method, i.e., the discriminative sparse coding (DSC) method [12], and the deep detail network (DDN) method [13], and three video-based methods, i.e., rain removal via matrix decomposition (VMD) [6], via stochastic analysis (VST) [11], and via temporal correlation/matrix low-rank completion (TCL) [14]. For VMD, the number of frames was set to $n = 11$ for our test videos as advised by the author. VST assumes the background is static throughout the video sequence, and thus distorts the background severely in our test videos with fast motion. To improve its performance, we split videos into batches of 10 frames and derain each batch individually.

4.2.1 Rain Streak Occlusion Precision Recall Rates

Rain fall introduces edges and textures over the background. Rain removal algorithms are expected to accurately detect and remove these high frequency contents, whilst preserving the edges and textures that belong to the background. To evaluate how much of the modifications from the derain algorithms contribute to only removing the rain pixels, we calculated the rain streak edge precision-recall (PR) curves. Absolute difference values were calculated between the derain output against the scene ground truth. Different threshold values were applied to retrieve a set of binary maps, which were next compared against the ground truth rain pixel map to calculate the PR rates. Higher PR value indicates better rain streak removal quality with less scene distortion.

Average PR curves for the two groups of testing scenes by different algorithms are shown in Fig. 10. We can make a few observations.

First, SPAC-CNN shows the best performance among all methods in both Group a and Group b. The performance of SPAC-CNN compared with SPAC-F1 and SPAC-Avg validates the efficiency of the CNN features $F_2$ and $F_3$, which correct mis-detected rain streaks (false negative rain detection), and compensate for misalignment blur (false positive rain detection).

Second, SPAC-RPCA is competitive with SPAC-F1 and outperforms SPAC-Avg for Group a, but falls behind these methods for Group b. RPCA models the scene background as low-rank, which gives tolerance to very small linear content shifts. It preserves background details better than SPAC-Avg when small camera motion is present. However, in Group b videos, faster camera motion and dynamic scenes reduces the similarity of scene content in Tensor $T_1$; this violates the low rank assumption required for RPCA, causing SPAC-RPCA to perform worse.

Third, for non-SPAC methods, video-based derain methods (TCL and VMD) perform better than image-based methods (DSC and DDN) for scenes in Group a. With slow camera motion, temporal correspondence can be more accurately established, which provides an advantage to video-based methods. However, with fast camera motion, temporal correspondence becomes less accurate, and the performance of video-based methods deteriorate relative to image-based methods.

Fourth, the SPAC methods are more robust to fast camera motion than other video-based rain removal methods. In particular, SPAC-RPCA is more robust than VMD and VST, although all three methods model the background as a low rank matrix. VST does not compensate for changes in the background. Hence, as seen in TABLE 3, it performs poorly in our test videos. VMD relies on global frame alignment to
compensate for camera motion; hence, it is robust towards slow camera motion. SPAC-RPCA aligns scene content at a superpixel granularity, which provides the best compensation for camera motion.

4.2.2 Scene Restoration PSNR/SSIM
We calculated the PSNR, and Structural Similarity (SSIM) of restored scenes after rain removal between different state-of-the-art methods against the ground truth, and the results are shown in TABLE 2.

| Data         | Group a Average | Group b Average |
|--------------|-----------------|-----------------|
| Tensor slice | F1 + F2 + F3    | F1 + F2 + F3    |
|              | (w/o F1)        | (w/o F2)        |
|               | 27.55 30.51 30.22 | 31.11 31.86 32.59 |

Under the SPAC framework, SPAC-RPCA shows better performance than SPAC-F1 and SPAC-Avg for Group a; however SPAC-F1 shows better performance for Group b. For both Groups a and b, SPAC-CNN is consistently 5 dB higher than SPAC-Avg, and 2 dB higher than SPAC-F1. SSIM of SPAC-CNN is also consistently highest among all SPAC methods. This further validates the efficiency of the CNN detail compensation network.

Video based methods (TCL and SPAC-CNN) outperform image-based methods for Group a data (around 2dB and 3dB higher respectively than DDN, 3dB and 5dB higher respectively than DSC). For Group b, image-based methods outperform TCL and VMD. However, SPAC-CNN still holds a 3dB advantage over DDN, 4dB over DSC.

4.2.3 Ablation Studies on CNN features
We evaluated the significance of different input features to the final derain PSNR over the two groups of testing data. Three baseline CNNs with different combinations of features as input were independently trained for this evaluation. As can be seen from the results in TABLE 2, a combination of all three features F1 + F2 + F3 provides the highest PSNR. F1 proves to be the most important feature. Visual inspection on the derain output show both F2 + F3 and F1 + F3 leave significant amount of un-removed rain. Comparing the last two columns, it can be seen that F3 works more efficiently for Group a than b (performance improvement of 0.9 dB and 0.6 dB, respectively), which is understandable since the high frequency features are better aligned for slow cameras, which leads to more accurate detail compensation.

4.3 Rain Removal Visual Comparisons
4.3.1 Visual Comparisons among SPAC Methods
Fig. 11 gives visual comparisons for the derain outputs from methods under the proposed SPAC framework. Several observations can be made:

1) SPAC-F1 shows smaller background distortions as compared with SPAC-RPCA and SPAC-Avg. However, since its output relies on the correctness of the rain mask, remnants of un-detected (false negative) rain streaks can often be found; for the falsely detected rain region (false positive), background distortion (blur) can be observed. These errors have been highlighted in yellow boxes in Fig. 11.

2) The overall quality of rain removal by SPAC-RPCA is good. The background is well preserved. There appear to be more, although less opaque, remnant rain streaks compared with SPAC-F1, as highlighted in green boxes. However, it gets worse when rain streaks are dense and opaque (last column).

3) SPAC-Avg is able to clear the rain streaks much cleaner than both SPAC-F1 and SPAC-RPCA, however it seriously blurs the background details.

4) SPAC-CNN provides the best visual performance among all SPAC methods. The rain is removed most cleanly, and the details are restored truthfully. The enhanced details have been highlighted in red boxes for comparison with SPAC-Avg.

4.3.2 Visual Comparisons with Other Recent Methods
We carried out visual comparison to compare the derain performance of recent rain removal algorithms. Fig. 12 shows the derain output from different competing methods for all synthetic testing data in Group a and b.

It is observed that rain can be much better removed by video-based methods. Image-based derain methods, i.e., DSC [12] and DDN [13], can only handle light and transparent rain occlusions. For those opaque rain streaks that cover a large area, they fail unavoidably. Temporal information proves to be critical in truthfully restoring the occluded details. Both VMD [6] and TCL [14] remove rain perfectly for Group a, however they create obvious blur when the camera motion is fast. The derain effect from SPAC-CNN is the most impressive for all methods.

Next, we compare these methods over real rain data. Fig. 13 shows the derain results over five sequences of real world rain videos. Two adjacent frames for the last two videos are shown to demonstrate the fast camera motion. It can be seen that, although the SPAC-CNN network has been trained over synthetic rain data, it generalizes well to real world rain as well. The advantage of SPAC-CNN is very obvious under heavy rain and fast camera motion.

4.3.3 Derain for Veiling & Accumulated Rain
As mentioned in Sec. 3.1.3, the averaging of T1 tensor slices is able to suppress both occluding rain streaks, and the out-of-focus rain drops near the camera (veiling rain). Fig. 14...
TABLE 3: Rain removal performance comparison between different methods in terms of scene reconstruction PSNR/SSIM. The best performance for each video sequence have been highlighted in underlined bold letters, and the second best have been highlighted with underlines.

| Camera Motion | Clip No. | Rain | DSC [12] | DDN [13] | VMD [6] | VST [11] | TCL [14] | SPAC | \( F_i \) | RPCA | Avg | CNN |
|---------------|----------|------|----------|----------|---------|----------|----------|------|-------|------|-----|-----|
| slow camera motion | a1 | 28.46/0.94 | 25.61/0.93 | 28.02/0.93 | 26.96/0.92 | 26.14/0.94 | 29.87/0.96 | 27.99/0.95 | 29.22/0.96 | 24.78/0.87 | 29.78/0.97 |
| | a2 | 28.09/0.95 | 27.11/0.95 | 27.38/0.95 | 24.80/0.93 | 24.03/0.83 | 29.01/0.96 | 29.27/0.95 | 29.95/0.96 | 26.34/0.89 | 30.09/0.96 |
| | a3 | 27.84/0.93 | 25.08/0.92 | 27.41/0.94 | 26.45/0.90 | 20.50/0.70 | 33.41/0.96 | 34.12/0.98 | 33.13/0.97 | 32.69/0.97 | 29.90/0.93 | 34.82/0.98 |
| | a4 | 31.48/0.95 | 28.82/0.95 | 32.47/0.97 | 29.55/0.94 | 29.78/0.97 | 32.64/0.89 | 29.01/0.94 | 29.15/0.95 | 24.72/0.85 | 29.75/0.96 |
| camera speed 20-30 km/h | b1 | 28.72/0.92 | 28.78/0.92 | 29.48/0.96 | 24.09/0.84 | 22.25/0.76 | 26.07/0.94 | 29.85/0.95 | 29.29/0.94 | 26.35/0.89 | 31.19/0.96 |
| | b2 | 29.49/0.90 | 29.58/0.92 | 30.23/0.95 | 25.81/0.89 | 25.13/0.79 | 32.41/0.97 | 31.26/0.96 | 31.52/0.96 | 28.83/0.93 | 34.05/0.98 |
| | b3 | 31.04/0.95 | 29.55/0.95 | 31.39/0.97 | 26.12/0.90 | 22.08/0.84 | 28.29/0.94 | 31.50/0.96 | 31.71/0.97 | 29.55/0.94 | 33.73/0.98 |
| | b4 | 27.99/0.92 | 29.10/0.93 | 29.83/0.96 | 25.90/0.88 | 25.63/0.80 | 30.38/0.95 | 31.92/0.95 | 31.25/0.96 | 28.85/0.92 | 33.79/0.97 |
| | b | 29.31/0.92 | 29.25/0.93 | 30.23/0.96 | 25.48/0.88 | 23.77/0.80 | 29.79/0.95 | 31.13/0.96 | 30.94/0.96 | 28.40/0.92 | 33.19/0.97 |

Fig. 11: Visual comparison for different rain removal methods under the SPAC framework for synthetic (column 1 to 4) and real world -rain (column 5 and 6). Electronic zoom-in is advised for better detail comparison.

shows two sequences of videos with veiling rain. The semi-transparent regional intensity fluctuations/distortions may not be apparent when observed in one frame; however, they are clearly visible and distracting in a video sequence. Fig. 14 shows zoom-in regions of 6 selected frames from the two videos. As can be seen, SPAC-CNN is able to suppress the fluctuations, and provide a stable clean background, even when the veiling rain covers a large portion of the frame.

As mentioned in Sec. 7 when the scene distance is large, the rain drops appear as aerosols and accumulate over the distance similar to that of fog and haze. After rain streak removal, dehazing can be applied as an additional step to improve vision contrast. Fig. 15 shows the dehazing effect using the non-local dehazer [19] over the output of SPAC-CNN. As can be seen, the accumulated rain effect can be effectively removed. The contrast and visibility of distant objects are well restored.
4.4 Execution Efficiency

We compared the average runtime between different methods for deraining a VGA resolution frame. We implemented \textit{SPAC-F}_1 and \textit{SPAC-Avg} in C++ with GPU acceleration. For further acceleration, global frame alignment was applied to bring the majority of scene content into alignment before accelerated template matching, where search was initially limited to a small volume, and later expanded to a larger volume if matches were not found. \textit{SPAC-RPCA} and \textit{SPAC-CNN}, on the other hand, were implemented in Matlab without the above acceleration methods. Results are shown in TABLE 4. As can be seen, \textit{SPAC-F}_1 and \textit{SPAC-Avg} are much faster than all other methods. \textit{SPAC-CNN} is much faster than all non-\textit{SPAC} video-based methods and comparable to the fastest image-based method \textit{DDN}.

TABLE 4: Execution time (in sec) comparison for different methods on deraining a single VGA resolution frame.

| Method          | Platform | Time  | Method | Platform | Time  |
|-----------------|----------|-------|--------|----------|-------|
| DSC-ICCV15 [12] | Matlab   | 236.3 | \textit{F}_1 | C++ | 0.2 |
| DDN-CVPR17 [13] | Matlab   | 0.9   | \textit{RPCA} | Matlab | 22.3 |
| VMD-CVPR17 [6]  | Matlab   | 119.0 | \textit{Avg} | C++ | 0.2 |
| TCL-TIP15 [14]  | Matlab   | 169.7 | \textit{CNN} | Matlab | 3.3 |

5 Conclusion and Discussion

5.1 Concluding Remarks

We have proposed a video-based rain removal framework based on SP content alignment and compensation. The proposed framework has been proven to be robust against fast camera motion, and can handle both opaque occluding rain streaks, and semi-transparent veiling rain that exhibits large areas of intensity fluctuations and distortions.
Fig. 13: Visual comparison between different rain removal methods on real world rain. Electronic zoom-in is advised for better detail comparison.

Fig. 14: Suppression of veiling rain. The first and third rows are zoom-in regions of 6 selected frames from two different video sequences. The second and fourth row are the derain outputs by SPAC-CNN.

As the basic processing unit of our proposed algorithms, although the SP template matching can only handle trans-
metrical motion, alignment errors caused by other types of motion such as rotation, scaling, and non-ridge transforms can be mitigated with global frame alignment before they are buffered (as shown in Fig. 3) [5]. Furthermore, these errors can be well modeled as sparse errors by SPAC-RPCA, and can be efficiently compensated by SPAC-CNN.

5.2 Future Work

The good performance of SPAC methods rely on the SP alignment accuracy. When camera moves even faster, SP search range \( n_s \) needs to be enlarged accordingly, which increases computation loads significantly. We have tested scenarios with camera speed going up to 50 km/h, the PSNR becomes lower since corresponding content are less likely be found within the search range. We believe a re-trained CNN with training data from such fast moving camera will help improve the performance. We plan to further investigate rain removal for even faster moving cameras (50 km/h+) in the future.

The framework of a SP-based alignment module plus a CNN compensation module has potential for various content alignment/registration applications under severe occlusion interference. We plan to further investigate these possibilities.

REFERENCES

[1] S. G. Narasimhan and S. K. Nayar, “Vision and the atmosphere,” International Journal of Computer Vision, vol. 48, no. 3, pp. 233–254, 2002.
[2] K. Garg and S. K. Nayar, “Vision and rain,” International Journal of Computer Vision, vol. 75, no. 1, pp. 3–27, 2007.
[3] C. H. Tan, J. Chen, and L.-P. Chau, “When does a camera see rain?” in IEEE International Conference on Computer Vision, vol. 2, Oct 2005, pp. 1067–1074.
[4] A. Tripathi and S. Mukhopadhyay, “Video post processing: low-latency spatiotemporal approach for detection and removal of rain,” IET Image Processing, vol. 6, no. 2, 2012.
[5] C.-H. Tan, J. Chen, and L.-P. Chau, “Dynamic scene rain removal for moving cameras,” in IEEE International Conference on Digital Signal Processing, IEEE, 2014, pp. 372–376.
[6] W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang, “Video desnowing and deraining based on matrix decomposition,” in IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1–4.
[7] K. Garg and S. K. Nayar, “Detection and removal of rain from videos,” in IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2004, pp. 528–533.
[8] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng, “Rain removal in video by combining temporal and chromatic properties,” in IEEE International Conference on Multimedia and Expo, July 2006, pp. 461–464.
[9] 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, vol. 93, no. 3, pp. 348–367, Jul 2011.
[10] J. Chen and L.-P. Chau, “A rain pixel recovery algorithm for videos with highly dynamic scenes,” IEEE Transactions on Image Processing, vol. 23, no. 3, pp. 1077–1104, 2014.
[11] W. Wei, L. Yi, X. Qie, Q. Zhao, D. Meng, and Z. Xu, “Should we encode rain streaks in video as deterministic or stochastic?” in IEEE International Conference on Computer Vision, Oct 2017, pp. 2535–2544.
[12] Y. Luo, Y. Xu, and H. Ji, “Removing rain from a single image via discriminative sparse coding,” in IEEE International Conference on Computer Vision, 2015, pp. 3397–3405.
[13] X. Fu, J. Huang, D. Z. Y. Huang, X. Ding, and J. Paisley, “Removing rain from single images via a deep detail network,” in IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[14] J. H. Kim, J. Y. Sim, and C. S. Kim, “Video deraining and desnowing using temporal correlation and low-rank matrix completion,” IEEE Transactions on Image Processing, vol. 24, no. 9, pp. 2658–2670, Sept 2015.
[15] J. Chen, C.-H. Tan, J. Hou, L.-P. Chau, and H. Li, “Robust video content alignment and compensation for rain removal in a CNN framework,” in IEEE Conference on Computer Vision and Pattern Recognition, 2018.
[16] J. Chen, C.-H. Tan, L.-P. Chau, and H. Li, “Rain Removal Based On Superpixel Spatial-Temporal Matching And Filtering,” Singapore Patent 10 201 703 975P, 2018.
[17] K. He, J. Sun, and X. Tang, “Single image haze removal using dark channel prior,” in IEEE Conference on Computer Vision and Pattern Recognition, June 2009, pp. 1956–1963.
[18] R. Fattal, “Dehazing using color-lines,” ACM Transactions on Graphics, vol. 34, no. 1, p. 13, 2014.
[19] D. Berman, S. Avidan et al., “Non-local image dehazing,” in IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1674–1682.
[20] S. You, R. T. Tan, R. Kawakami, Y. Mukaigawa, and K. Ikeuchi, “Adherent raindrop modeling, detectionand removal in video,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 9, pp. 1721–1733, Sept 2016.
[21] 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, Dec. 2013, pp. 633–640.
[22] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, “Automatic single-image-based rain streaks removal via image decomposition,” IEEE Transactions on Image Processing, vol. 21, no. 4, pp. 1742–1755, Apr. 2012.

[23] D.-A. Huang, L.-W. Kang, Y.-C. Wang, and C.-W. Lin, “Self-Learning Based Image Decomposition With Applications to Single Image Denoising,” IEEE Transactions on Multimedia, vol. 16, no. 1, pp. 83–93, Jan. 2014.

[24] S.-H. Sun, S.-P. Fan, and Y.-C. Wang, “Exploiting image structural similarity for single image rain removal,” in 2014 IEEE International Conference on Image Processing (ICIP), Oct. 2014, pp. 4482–4486.

[25] D.-Y. Chen, C.-C. Chen, and L.-W. Kang, “Visual depth guided coherent image rain streaks removal using sparse coding,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 24, no. 8, pp. 1430–1455, Aug. 2014.

[26] 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), Dec. 2013, pp. 1968–1975.

[27] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, “Rain streak removal using layer priors,” in IEEE Conference on Computer Vision and Pattern Recognition, June 2016, pp. 2736–2744.

[28] ——, “Single Image Rain Streak Decomposition Using Layer Priors,” IEEE Transactions on Image Processing, vol. 26, no. 8, pp. 3874–3885, Aug. 2017.

[29] J.-H. Kim, C. Lee, J.-Y. Sim, and C.-S. Kim, “Single-image deraining using an adaptive nonlocal means filter,” in IEEE International Conference on Image Processing, Sep. 2013, pp. 914–917.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105.

[31] J. Kim, J. K. Lee, and K. M. Lee, “Accurate image super-resolution using very deep convolutional networks,” in IEEE Conference on Computer Vision and Pattern Recognition, June 2016, pp. 1646–1654.

[32] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising,” IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3142–3155, July 2017.

[33] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, “Deep joint rain detection and removal from a single image,” in IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1357–1366.

[34] T. X. Jiang, T. Z. Huang, X. L. Zhao, L. J. Deng, and Y. Wang, “A Novel Tensor-Based Video Rain Streaks Removal Approach via Utilizing Discriminatively Intrinsic Priors,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jul. 2017, pp. 2818–2827.

[35] P. C. Barnum, S. Narasimhan, and T. Kanade, “Analysis of rain and snow in frequency space,” International Journal of Computer Vision, vol. 86, no. 2, p. 236, Jan 2009.

[36] V. Santhanaseelan and V. K. Asari, “Utilizing local phase information to remove rain from video,” International Journal of Computer Vision, vol. 112, no. 1, pp. 71–89, 2015.

[37] 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, vol. 26, pp. 317–328, Jan. 2015.

[38] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” SIAM review, vol. 51, no. 3, pp. 455–500, 2009.

[39] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, “RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pp. 2233–2246, Nov 2012.

[40] T. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “SPEED-UP Robust Features (SURF),” Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346 – 359, 2008.

[41] P. H. Torr and A. Zisserman, “MLESAC: a new robust estimator with application to estimating image geometry,” Computer Vision and Image Understanding, vol. 78, no. 1, pp. 138–156, 2000.

[42] A. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pp. 2274–2282, 2012.

[43] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool, SEEDS: Superpixels Extracted via Energy-Driven Sampling. Springer Berlin Heidelberg, 2012, pp. 13–26.