

Convolutional Sparse and Low-Rank Coding-Based Rain Streak Removal

He Zhang and Vishal M. Patel
Department of Electrical and Computer Engineering
Rutgers University, Piscataway, NJ 08854
{he.zhang92, vishal.m.patel}@rutgers.edu

Abstract

We propose a novel Convolutional Coding-based Rain Removal (CCRR) algorithm for automatically removing rain streaks from a single rainy image. Our method first learns a set of generic sparsity-based and low-rank representation-based convolutional filters for efficiently representing background clear image and rain streaks, respectively. To this end, we first develop a new method for learning a set of convolutional low-rank filters. Then, using these learned filter, we propose an optimization problem to decompose a rainy image into a clear background image and a rain streak image. By working directly on the whole image, the proposed rain streak removal algorithm does not need to divide the image into overlapping patches for learning local dictionaries. Extensive experiments on synthetic and real images show that the proposed method performs favorably compared to state-of-the-art rain streak removal algorithms.

1. Introduction

Many computer vision algorithms such as detection, tracking and classification are specifically designed for images and videos that are collected in controlled and constrained environmental conditions. However, in many applications such as self driving cars and aerial imaging, one has to process images and videos that collected in outdoor unconstrained environments containing undesirable artifacts such as rain, snow, and fog. The performance of many computer vision algorithms often degrades when they are presented with the outdoor images containing some of these artifacts. As a result, removing these artifacts automatically before applying them to various computer vision algorithms can improve their performances significantly.

Various methods have been developed in the literature for removing these artifacts. Some of them include automatic haze removal [9, 22, 14, 18], fog removal [17], snow removal [33], and rain streak removal [12, 5, 4, 16, 15, 13, 6]. In this paper, we focus on the problem of automatically



(a) Input (b) Ground Truth (c) Rain streak

Figure 1: Rain streak removal from a single image. A rainy image (a) can be viewed as the superposition of clean background image (b) and rain streak image (c).

removing rain streaks from a single image. Rain streaks often reduce the visibility due to their scattering and blurring artifacts. Automatic de-raining or rain streak removal is a difficult inverse problem because we do not know the location and the statistics of the rain streaks in the given image. The amount and location of rain streaks often change depending on how hard it is raining.

One can model the observed rainy image as the superposition of two images - one corresponding to rain streak and the other corresponding to clear background (see Figure 1) [15, 4, 12]. Let $\mathbf{y} \in \mathbb{R}^{MN}$, $\mathbf{y}_c \in \mathbb{R}^{MN}$ and $\mathbf{y}_r \in \mathbb{R}^{MN}$ be the lexicographically ordered vectors corresponding to $M \times N$ rainy image (Y), clear background image (Y_c) and rain streak image (Y_r), respectively. Then, the input rainy image can be expressed as

$$\mathbf{y} = \mathbf{y}_c + \mathbf{y}_r. \quad (1)$$

Given \mathbf{y} , the goal of rain streak removal is to decompose it into \mathbf{y}_c and \mathbf{y}_r and the recovered \mathbf{y}_c represents the de-rained image.

Assuming that \mathbf{y}_c and \mathbf{y}_r can be sparsely represented in appropriate dictionaries, several recent works have proposed sparsity-based optimization algorithms for separating rain streaks and background images from the rainy images [12, 11, 16, 25]. For instance, [12] proposed a sparsity-based method to cluster the learned dictionary atoms into two groups. Then using these two groups of atoms, they

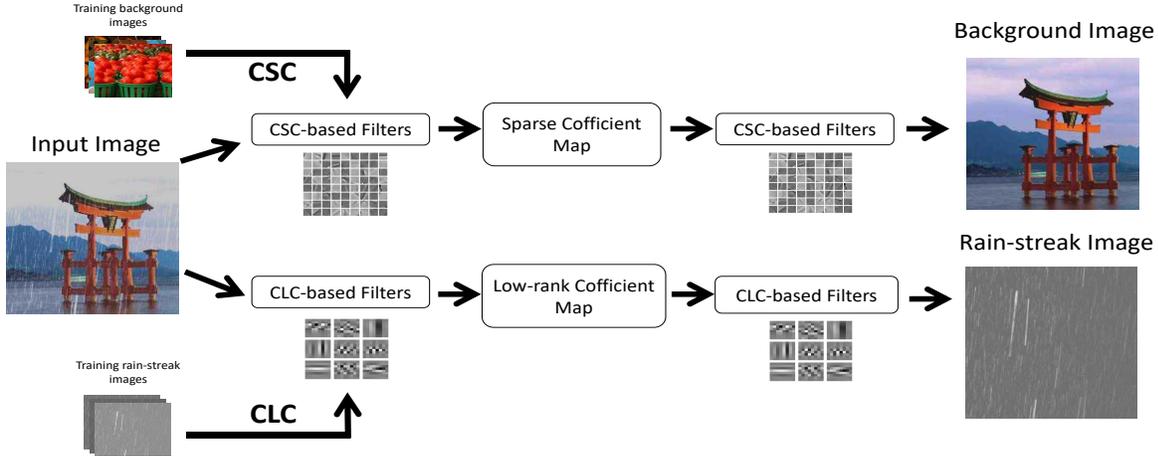


Figure 2: An overview of the proposed convolutional coding-based rain streak removal algorithm.

develop a method to separate a rainy image into rain streaks and clear images. One of the limitations of this method is that the recovered clean image tends to be smooth with blurry edges. Another discriminative sparse coding-based de-raining method was proposed in [16] in which mutual exclusivity property is enforced to find the sparse codes for separation. In [11], unsupervised clustering was applied on the observed dictionary atoms via affinity propagation to identify the image dependent components for de-raining. A nonlocal means filtering framework was proposed in [13] for filtering elliptical shapes corresponding to rain drops in the given image. Motivated by the low-rank property of textures [23, 19], a low-rank representation-based framework for rain streak removal was proposed in [4]. Gaussian Mixture Model (GMM) based [15] as well as Convolutional Neural Network (CNN) based [5] methods have also been proposed for de-raining in the literature. Several authors have made use of the temporal information in a video for de-raining [32, 7, 6].

One of the limitations of some of these approaches such as [12, 4] is that they are patch-based and they do not specifically use the global structure of the background image. As a result, the de-rained image often tends to contain components from the background image or the background image often contains rain components. Furthermore, sparse dictionary-based rain streak removal methods use image patches to learn local dictionaries. As a result, they often contain shifted versions of the same features [2]. To deal with this, Convolutional Sparse Coding (CSC) methods have been introduced in which shift invariance is directly modeled in the objective [29, 2, 10, 28]. CSC has been successfully applied in various image processing and computer vision applications [30, 20, 8, 31].

In this paper, we present a CSC and Convolutional Low-Rank Coding (CLC) based method for rain streak removal from a single image. We first learn a set of CSC and CLC

filters to efficiently represent the background image and rain streaks, respectively. Then, using the learned filters, we develop an image separation algorithm based on sparse and low-rank coding. Figure 2 gives an overview of the proposed Convolutional Coding-based Rain Removal (CCRR) method.

This paper makes the following contributions.

1. We present an optimization framework for CLC for efficiently representing low-rank rain streaks.
2. CCRR is proposed in which pre-trained CSC and CLC filters are used to efficiently represent background image and rain streaks, respectively. Using these filters, we propose an image separation method based on sparse and low-rank coding for rain streak removal.
3. We develop alternating direction method of multipliers (ADMM) based optimization frameworks [1] for solving the proposed CLC and CCRR algorithms.

Rest of the paper is organized as follows. In Section 2, we give a brief introduction to CSC and formulate the proposed CCRR method. Details of the CCRR optimization are given in Section 3. Experimental results are presented in Section 4. Section 5 concludes the paper with a brief summary and discussion.

2. Background and Problem Formulation

In this section, we give a brief background on CSC and formulate the proposed convolutional coding-based rain streak removal problem.

2.1. Convolutional Sparse Coding

In CSC, given a set of M training samples $\{\mathbf{y}_m\}_{m=1}^M$, the objective is to learn a set of convolutional filters $\{\mathbf{d}_k\}_{k=1}^K$

by solving the following optimization problem

$$\begin{aligned} \arg \min_{\mathbf{d}, \mathbf{x}} \quad & \frac{1}{2} \sum_{m=1}^M \left\| \mathbf{y}_m - \sum_{k=1}^K \mathbf{d}_k * \mathbf{x}_{m,k} \right\|_2^2 \\ & \lambda \sum_{m=1}^M \sum_{k=1}^K \|\mathbf{x}_{m,k}\|_1 \end{aligned} \quad (2)$$

subject to $\|\mathbf{d}_k\|_2^2 \leq 1 \quad \forall k \in \{1, \dots, K\}$,

where $\mathbf{x}_{m,k}$ are the sparse coefficients that approximate the data \mathbf{y}_m when convolved with the corresponding filters \mathbf{d}_k of fixed support and for an N -dimensional vector \mathbf{x} , $\|\cdot\|_q$ denotes the q -norm, $0 < q < \infty$, defined as $\|\mathbf{x}\|_q = \left(\sum_{i=1}^N |x_i|^q \right)^{\frac{1}{q}}$. Here, $*$ represents the 2-D convolution operator and λ is a positive regularization parameter. Several methods have been proposed in the literature for solving the above optimization problem [2, 10, 28, 27]. In particular, [28], [27] developed an efficient method that jointly uses the space and Fourier domains to solve the CSC problem. In this paper, we adapt the method proposed in [28] for learning the convolutional filters due to its simplicity and efficiency.

2.2. Convolutional Coding-based Rain Removal

In order to separate the rain streaks and the background image from the mixture model (1), we need efficient representations for rainy component and the background image. Since rain streaks are texture like, they are inherently low-rank in nature. In fact, this assumption has been used in [4] for de-raining. In order to learn a set of low-rank filters for efficiently representing rain streaks, we propose the following CLC problem

$$\begin{aligned} \arg \min_{\mathbf{d}, \mathbf{x}} \quad & \frac{1}{2} \sum_{m=1}^M \left\| \mathbf{y}_m - \sum_{k=1}^K \mathbf{d}_k * \mathbf{x}_{m,k} \right\|_2^2 \\ & \lambda_l \sum_{m=1}^M \sum_{k=1}^K \|\mathbf{x}_{m,k}\|_* \end{aligned} \quad (3)$$

subject to $\|\mathbf{d}_k\|_2^2 \leq 1 \quad \forall k \in \{1, \dots, K\}$,

where $\|\cdot\|_*$ is nuclear norm, representing the sum of singular values and λ_l is a positive regularization parameter.

Assume that we have learned a set of convolutional sparsity-based filters $\{\mathbf{d}_{c,k}\}$ using CSC to sparsely represent the clear background part and another set of convolutional low-rank-based filters $\{\mathbf{d}_{r,k}\}$ using CLC to efficiently represent the rain streaks. That is, we have learned $\{\mathbf{d}_{c,k}\}_{k=1}^{K_c}$ and $\{\mathbf{d}_{r,k}\}_{k=1}^{K_r}$ such that $\mathbf{y}_c = \sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k}$ and $\mathbf{y}_r = \sum_{k=1}^{K_r} \mathbf{d}_{r,k} * \mathbf{x}_{r,k}$, where $\mathbf{x}_{c,k}$ are the sparse coefficients and $\mathbf{x}_{r,k}$ are the low-rank coefficients that approximate \mathbf{y}_c and \mathbf{y}_r when convolved with the filters $\mathbf{d}_{c,k}$ and $\mathbf{d}_{r,k}$, respectively. Then, we propose to estimate the clear background and rain components via $\mathbf{x}_{c,k}$ and $\mathbf{x}_{r,k}$, respec-

tively by solving the following CCRR optimization problem

$$\begin{aligned} \hat{\mathbf{x}}_{c,k}, \hat{\mathbf{x}}_{r,k} = \arg \min_{\mathbf{x}_{c,k}, \mathbf{x}_{r,k}} \quad & \frac{1}{2} \left\| \mathbf{y} - \sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k} - \sum_{k=1}^{K_r} \mathbf{d}_{r,k} * \mathbf{x}_{r,k} \right\|_2^2 \\ & + \lambda_c \sum_{k=1}^{K_c} \|\mathbf{x}_{c,k}\|_1 + \lambda_r \sum_{k=1}^{K_r} \|\mathbf{x}_{r,k}\|_* \\ & + \beta TV \left(\sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k} \right), \end{aligned} \quad (4)$$

where β , λ_r and λ_c are positive regularization parameters and TV is the total variation (i.e. sum of the absolute variations in the image). Note that in CCRR, we enforce sparsity constraint on the coefficients corresponding to the background image and low-rank constraint on the coefficients corresponding to the rain streaks. Once, $\mathbf{x}_{c,k}$ and $\mathbf{x}_{r,k}$ are estimated, the two components can be obtained by $\hat{\mathbf{y}}_c = \sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \hat{\mathbf{x}}_{c,k}$ and $\hat{\mathbf{y}}_r = \sum_{k=1}^{K_r} \mathbf{d}_{r,k} * \hat{\mathbf{x}}_{r,k}$, where $\hat{\mathbf{y}}_c$ represents the de-rained image.

3. Optimization

In this section, we derive the framework for solving the proposed CLC and CCRR optimization problems.

3.1. CLC

The problem (3) can be solved by iteratively updating \mathbf{d}_k and $\mathbf{x}_{m,k}$, as it is a bi-convex optimization problem. The updating procedure is as follows:

3.1.1 Fix $\mathbf{x}_{m,k}$ and update \mathbf{d}_k

We solve the following optimization problem for updating each filter

$$\begin{aligned} \arg \min_{\mathbf{d}_k} \quad & \frac{1}{2} \sum_{m=1}^M \left\| \mathbf{y}_m - \sum_{k=1}^K \mathbf{d}_k * \mathbf{x}_{m,k} \right\|_2^2 \\ \text{subject to} \quad & \|\mathbf{d}_k\|_2 \leq 1, \quad \forall k. \end{aligned} \quad (5)$$

We can regard the constrains $\|\mathbf{d}_k\|_2 \leq 1$ as post-processing after each iteration. Then, (5) can be rewritten as

$$\arg \min_{\mathbf{d}_k} \quad \frac{1}{2} \sum_{m=1}^M \left\| \mathbf{y}_m - \sum_{k=1}^K \mathbf{d}_k * \mathbf{x}_{m,k} \right\|_2^2. \quad (6)$$

To solve (6) in the DFT domain, we zero pad \mathbf{d}_k so that it has the same spatial support as $\mathbf{x}_{m,k}$. We form another optimization problem (7) that can directly include the zero-padding and normalization procedure for \mathbf{d}_k in the objective [28] as

$$\begin{aligned} \arg \min_{\mathbf{d}_k, \mathbf{g}_k} \quad & \frac{1}{2} \sum_{m=1}^M \left\| \mathbf{y}_m - \sum_{k=1}^K \mathbf{d}_k * \mathbf{x}_{m,k} \right\|_2^2 + \sum_{k=1}^K l_{C_{zp}}(\mathbf{g}_k) \\ \text{subject to} \quad & \mathbf{d}_k - \mathbf{g}_k = 0 \quad \forall k, \end{aligned} \quad (7)$$

where $l_{C_{zp}}$ is the indicator function of the constraint set C_{zp} ¹. The iterative update methods for solving (7) via scaled form of ADMM are as follows

$$\mathbf{d}_k^{(j+1)} = \arg \min_{\mathbf{d}_k} \frac{1}{2} \sum_{m=1}^M \|\mathbf{y}_m - \sum_{k=1}^K \mathbf{d}_k * \mathbf{x}_{m,k}\|_2^2 + \frac{\sigma}{2} \sum_{k=1}^K \|\mathbf{d}_k - \mathbf{g}_k^{(j)} + \mathbf{q}_k^{(j)}\|_2^2, \quad (9)$$

$$\mathbf{g}_k^{(j+1)} = \arg \min_{\mathbf{g}_k} \sum_{k=1}^K l_{C_{zp}}(\mathbf{g}_k) + \frac{\sigma}{2} \sum_{k=1}^K \|\mathbf{d}_k^{(j+1)} - \mathbf{g}_k + \mathbf{q}_k^{(j)}\|_2^2, \quad (10)$$

$$\mathbf{q}_k^{(j+1)} = \mathbf{q}_k^{(j)} + \mathbf{d}_k^{(j+1)} - \mathbf{g}_k^{(j+1)}, \quad (11)$$

where \mathbf{q} is the scaled dual variable. The optimization problem (9) can be solved using the DFT-based method proposed in [28], and (10) can be solved using a proximal algorithm [21].

3.1.2 Fix \mathbf{d}_k and update $\mathbf{x}_{m,k}$

We rewrite (3) as

$$\arg \min_{\mathbf{x}_{m,k}, \mathbf{z}_{m,k}} \frac{1}{2} \sum_{m=1}^M \|\mathbf{y}_m - \sum_{k=1}^K \mathbf{d}_k * \mathbf{x}_{m,k}\|_2^2 + \lambda_l \sum_{m=1}^M \sum_{k=1}^K \|\mathbf{z}_{m,k}\|_* \quad (12)$$

subject to $\mathbf{x}_{m,k} - \mathbf{z}_{m,k} = 0, \forall k$.

Then, the iterative update rules for solving (12) are as follows

$$\mathbf{x}_{m,k}^{(j+1)} = \arg \min_{\mathbf{x}_{m,k}} \frac{1}{2} \|\mathbf{y}_m - \sum_{k=1}^K \mathbf{d}_k * \mathbf{x}_{m,k}\|_2^2 + \frac{\rho}{2} \sum_{k=1}^K \|\mathbf{x}_{m,k} - \mathbf{z}_{m,k}^{(j)} + \mathbf{u}_{m,k}^{(j)}\|_2^2, \quad (13)$$

$$\mathbf{z}_{m,k}^{(j+1)} = \arg \min_{\mathbf{z}_{m,k}} \lambda_l \sum_{k=1}^K \|\mathbf{z}_{m,k}\|_* + \frac{\rho}{2} \sum_{k=1}^K \|\mathbf{x}_{m,k}^{(j+1)} - \mathbf{z}_{m,k} + \mathbf{u}_{m,k}^{(j)}\|_2^2, \quad (14)$$

$$\mathbf{u}_{m,k}^{(j+1)} = \mathbf{u}_{m,k}^{(j)} + \mathbf{x}_{m,k}^{(j+1)} - \mathbf{z}_{m,k}^{(j+1)}. \quad (15)$$

¹ $l_C()$ is defined as

$$l_C(p) = \begin{cases} 0, & \text{if } p \in C \\ \infty, & \text{if } p \notin C. \end{cases} \quad (8)$$

Problem (13) can be solved using the optimization method proposed in [28] and (14) can be solved using Singular Value Thresholding (SVT) [3].

3.2. CCRR

If we discard the TV part in (4), then the resulting optimization problem can be solved iteratively over $\mathbf{x}_{c,k}$ and $\mathbf{x}_{r,k}$.

3.2.1 Update step for $\mathbf{x}_{c,k}$

When $\mathbf{x}_{r,k}$ is fixed, we need to solve the following problem to obtain the sparse coefficients $\mathbf{x}_{c,k}$

$$\hat{\mathbf{x}}_{c,k} = \arg \min_{\mathbf{x}_{c,k}} \frac{1}{2} \left\| \mathbf{y} - \sum_{k=1}^{K_r} \mathbf{d}_{r,k} * \mathbf{x}_{r,k} - \sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k} \right\|_2^2 + \lambda_c \sum_{k=1}^{K_c} \|\mathbf{x}_{c,k}\|_1. \quad (16)$$

This problem can be solved using the DFT-based ADMM method [28].

3.2.2 Update step for $\mathbf{x}_{r,k}$

For a fixed $\mathbf{x}_{c,k}$, we have to solve the following problem to obtain $\mathbf{x}_{r,k}$

$$\hat{\mathbf{x}}_{r,k} = \arg \min_{\mathbf{x}_{r,k}} \frac{1}{2} \left\| \mathbf{y} - \sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k} - \sum_{k=1}^{K_r} \mathbf{d}_{r,k} * \mathbf{x}_{r,k} \right\|_2^2 + \lambda_r \sum_{k=1}^{K_r} \|\mathbf{x}_{r,k}\|_*, \quad (17)$$

This problem is very similar to the sub-problem that we solve in CLC for finding the low-rank coefficients when \mathbf{d}_k are fixed. Let $\mathbf{y}_p = \mathbf{y} - \sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k}$. Then (17) can be rewritten as

$$\hat{\mathbf{x}}_{r,k} = \arg \min_{\mathbf{x}_{r,k}} \frac{1}{2} \left\| \mathbf{y}_p - \sum_{k=1}^{K_r} \mathbf{d}_{r,k} * \mathbf{x}_{r,k} \right\|_2^2 + \lambda_r \sum_{k=1}^{K_r} \|\mathbf{x}_{r,k}\|_*,$$

which can be solved using the optimization procedure described in the previous subsection for CLC.

Finally, the TV correction is applied only on the background rain-free part to control the edges in the clear image. The overall CCRR algorithm for rain streak removal is summarized in Algorithm 1, where L is the total iteration number and i is the iteration index.

4. Experimental Results

In this section, we present the results of our proposed CCRR algorithm for single image de-raining on both gray-scale and color images. We compare the performance of our method with that of four state-of-the-art single image de-raining methods - sparse dictionary-based method

Algorithm 1: The CCRR Algorithm for Rain Removal.

```
1 Input:  $\{\mathbf{d}_{c,k}\}_{k=1}^{K_c}, \{\mathbf{d}_{r,k}\}_{k=1}^{K_r}, \mathbf{y}, \lambda_c, \lambda_r, L$ 
2 Initialization
3 for  $i = 1 : L$ 
4     Obtain  $\mathbf{x}_{c,k}$  by solving (4) when fixing  $\mathbf{x}_{r,k}$ .
5     Obtain  $\mathbf{y}_c$  by applying the TV correction [24].
6     Use  $\mathbf{y}_c$  to replace  $\sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \hat{\mathbf{x}}_{c,k}$  in (4).
7     Obtain  $\hat{\mathbf{x}}_{r,k}$  by solving (4) when fixing  $\mathbf{x}_{c,k}$ .
8 end for
9  $\hat{\mathbf{y}}_r = \sum_{k=1}^{K_r} \mathbf{d}_{r,k} * \hat{\mathbf{x}}_{r,k}$ ,
10  $\hat{\mathbf{y}}_c = \mathbf{y} - \hat{\mathbf{y}}_r$ ;
11 Output:  $\hat{\mathbf{y}}_c, \hat{\mathbf{y}}_r$ 
```

(Auto-SP) [12], discriminative sparse coding-based method (Dis-SP) [16], low-rank representation-based method (Low-rank) [4] and a CNN-based method (CNN) [5]. In these experiments, we use Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [26] to measure the performance of the routines tested.

Sample training images shown in Figure 3 (a) are used to learn the convolutional sparse filters $\{\mathbf{d}_{c,k}\}$ using CSC. Similarly, some training images in the Figure 3 (b) are utilized to learn the convolutional low-rank filters $\{\mathbf{d}_{r,k}\}$ using CLC. The corresponding CSC and CLC learned filters are shown in Figure 3 (c) and (d), respectively. The size of each CSC filter is set equal to 8×8 and the size of each CLC filter is set equal to 6×6 for all experiments. From Figure 3 (d), one can see that these filters are oscillatory in nature and they do a good job in capturing the rain texture patterns. These filters can capture the low-rank structure found in the rain streaks. Similarly, from Figure 3 (c), we observe that the learned filters look similar to those found in a Gabor or curvelet dictionary. They capture domain specific information found in natural images such as edges and contours.

All testing images are excluded from the training procedure. For the gray-scale images, the parameters λ_c and λ_r are set equal to $\max(0.55 - 0.090 * i, 0.001)$ and $\max(5.20 - 0.90 * i, 0.05)$, respectively. For the color images, the parameters are set as $\lambda_c = \max(1.35 - 0.435 * i, 0.001)$ and $\lambda_r = \max(5.30 - 0.72 * i, 0.82)$. The total iteration number L is set equal to to 6 for all experiments.

4.1. Rain Removal from Gray-scale Image

In the first set of experiments, we evaluate the quantitative performance of different methods on the two synthetic gray-scale images released by Kang *et al.* in [12]. These synthetic rainy images are shown in Figure 5. The performance of different de-raining methods in terms of PSNR and SSIM is tabulated in Table 1. As can be seen from this table, on average our method performs favorably over some of the compared methods.

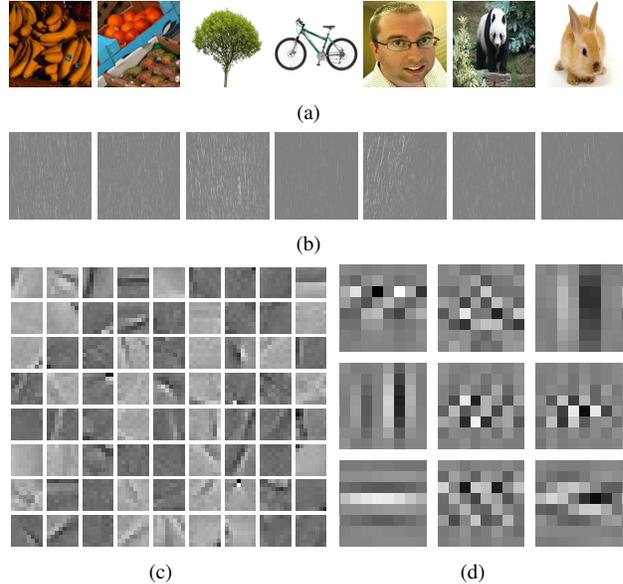


Figure 3: (a) Training non-rain images used for learning a set of sparsity-based filters $\{\mathbf{d}_{c,k}\}$. (b) Rain-streak images used for learning a set of low-rank filters $\{\mathbf{d}_{r,k}\}$. (c) Learned non-rain filters $\{\mathbf{d}_{c,k}\}$. (d) Learned rain-streak filters $\{\mathbf{d}_{r,k}\}$.

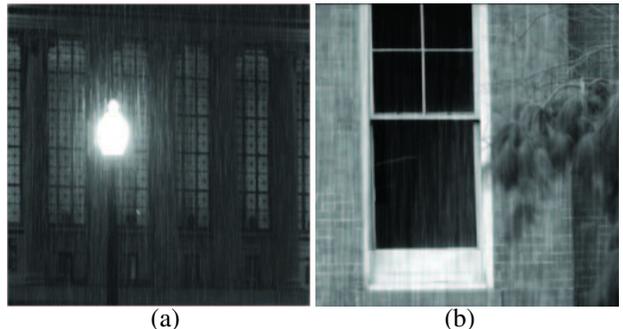


Figure 5: Synthetic gray-scale rainy images.

		Rainy	CCRR	Auto-SP [12]	Low-rank [4]	Dis-sp [16]	CNN [5]
SSIM	Fig 5(a)	0.6602	0.7699	0.7410	0.7641	0.6738	0.7751
	Fig 5(b)	0.8579	0.8939	0.8654	0.8905	0.8774	0.8510
PSNR (dB)	Fig 5(a)	24.75	25.78	24.78	25.42	25.61	24.82
	Fig 5(b)	24.44	25.17	24.49	24.58	25.01	23.62

Table 1: Results on two synthetic gray-scale images.

In the second set of experiments with the gray-scale images, we use a real rainy gray-scale image (shown in the first row of Figure 4) and visually inspect the performance of different de-raining methods by displaying the separated clear background images (shown in the second row of Figure 4) and rain streak images (shown in the third row of Figure 4) corresponding to different methods. From the third row, we can observe that our method can capture more rain streaks and less of other structures, demonstrating the advantage of using convolutional low-rank filters over just using sparsity

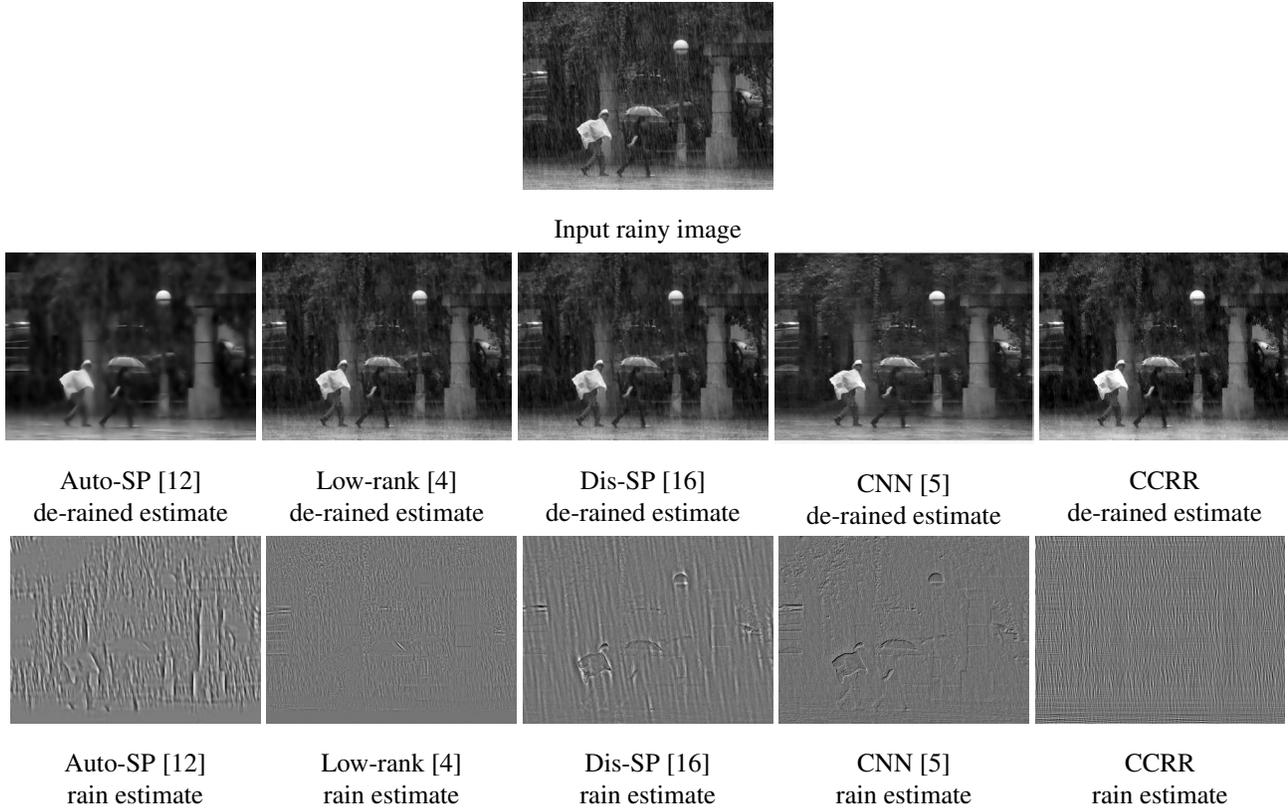


Figure 4: Rain-streak removal results on a real gray-scale image.

as prior for the rain component. We also observe that the recovered rain-streak components from Dis-SP, Auto-SP and CNN methods capture more information corresponding to the background non-rain image.

4.2. Rain Removal from Color Images

Synthetic Images. We used two color rainy images released by [12] with its ground truth to measure the de-raining performance of different methods. The results are shown in Fig 6. It can be observed that our method outperforms all the other three methods quantitatively as well as qualitatively. For example, from this figure, we see that the Auto-SP method [12] tends to smooth the de-rained part when removing the rain component, while the low-rank method [4] fails to capture some rain components in the background image. Even though the Dis-SP and CNN methods have a very competitive visual quality, some rainy components still remain in the de-rained image for both of these methods. Furthermore, Dis-SP tend to enhance the contrast of some details such as the face shown in the first row in Figure 6 and the CNN-based rain removal method has very poor quantitative performance. Similar visual and quantitative results are also observed in the second and the third row of Figure 6.

Real Images. We also evaluated the performance of our proposed method on many real images downloaded from the Internet. The de-rained results for all the methods and their corresponding input rainy images are shown in Figure 7. The first row shows the input rainy images. Results of Auto-SP [12], Low-rank-based method [4], Dis-SP [16], CNN-based method [5] and our CCRR method are shown in the second to fourth rows, respectively. From these de-rained images, we observe that the Auto-SP and Low-rank methods tend to smooth the details in the de-rained images even though they can remove a lot of rain streaks. This can be seen by observing the head part of the athletes in the third column of this figure. The de-rained images from the Dis-SP method still contains a lot of rain streaks. In general, the CNN-based method achieves very good visual quality, however, it fails to tackle heavy rain conditions, as can be seen by comparing the results in the second and last columns of Figure 7. Our proposed CCRR method can preserve the details such as edges and contours while removing the low-rank rain streaks. This experiment clearly demonstrates the significance of the proposed method in removing rain streaks from real-world rainy images under a variety of different background and conditions.



Ground Truth	Input	Auto-SP [12]	Low-rank [4]	Dis-SP [16]	CNN [5]	CCRR
	SSIM:0.7571 PSNR: 18.75	SSIM:0.7853 PSNR: 21.23	SSIM:0.8153 PSNR: 22.74	SSIM:0.8419 PSNR: 24.84	SSIM:0.7584 PSNR: 20.00	SSIM: 0.8662 PSNR: 25.56



Ground Truth	Input	Auto-SP [12]
	SSIM:0.6959 PSNR: 20.21	SSIM:0.8181 PSNR: 21.74



Low-rank [4]	Dis-SP [16]	CNN [5]	CCRR
SSIM:0.8215 PSNR: 21.87	SSIM:0.7915 PSNR: 24.15	SSIM:0.7326 PSNR: 21.29	SSIM: 0.8526 PSNR: 26.05

Figure 6: Rain-streak removal results on two synthetic color images. We compare the performance of our proposed CCRR method with other three methods: Auto-SP [12], Low-rank [4] and Dis-SP [16].

5. Conclusion

We proposed the CCRR algorithm for removing rain streaks from a given rainy image. Our method entails learning sparsity-based and low-rank representation-based filters directly from training examples. Using these learned filters, we proposed an optimization framework for de-raining. Various experiments showed the significance of our CCRR de-raining method over several recent state-of-the-art de-raining methods.

In the future, we will investigate the possibility of developing convolutional sparse and low-rank coding-based methods for haze and fog removal.

$$\begin{aligned}
 \hat{\mathbf{x}}_{c,k}, \hat{\mathbf{x}}_{r,k} = \arg \min_{\mathbf{x}_{c,k}, \mathbf{x}_{r,k}} \frac{1}{2} & \left\| \mathbf{y} - \sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k} - \sum_{k=1}^{K_r} \mathbf{d}_{r,k} * \mathbf{x}_{r,k} \right\|_2^2 \\
 & + \lambda_c \sum_{k=1}^{K_c} \|\mathbf{x}_{c,k}\|_1 + \lambda_r \sum_{k=1}^{K_r} \|\mathbf{x}_{r,k}\|_* \\
 & + \beta TV \left(\sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k} \right),
 \end{aligned}$$

(18)



Input



Auto-SP [12]



Low-rank [4]



Dis-SP [16]



CNN [5]



CCRR

Figure 7: Rain-streak removal results on three real images. We compare the performance of our proposed CCRR method with the other three methods: Auto-SP [12], Low-rank [4], Dis-SP [16] and CNN [5].

References

- [1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.
- [2] H. Bristow, A. Eriksson, and S. Lucey. Fast convolutional sparse coding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 391–398, 2013.
- [3] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.
- [4] Y.-L. Chen and C.-T. Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *IEEE International Conference on Computer Vision*, pages 1968–1975, 2013.
- [5] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley. Clearing the Skies: A deep network architecture for single-image rain removal. *ArXiv e-prints*, Sept. 2016.
- [6] K. Garg and S. K. Nayar. Detection and removal of rain from videos. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–528. IEEE.
- [7] K. Garg and S. K. Nayar. Vision and rain. *International Journal of Computer Vision*, 75(1):3–27, 2007.
- [8] S. Gu, W. Zuo, Q. Xie, D. Meng, X. Feng, and L. Zhang. Convolutional sparse coding for image super-resolution. In *IEEE International Conference on Computer Vision*, pages 1823–1831, Dec 2015.
- [9] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2341–2353, 2011.
- [10] F. Heide, W. Heidrich, and G. Wetzstein. Fast and flexible convolutional sparse coding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5135–5143. IEEE, 2015.
- [11] D.-A. Huang, L.-W. Kang, Y.-C. F. Wang, and C.-W. Lin. Self-learning based image decomposition with applications to single image denoising. *IEEE Transactions on multimedia*, 16(1):83–93, 2014.
- [12] 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*, 21(4):1742–1755, 2012.
- [13] J.-H. Kim, C. Lee, J.-Y. Sim, and C.-S. Kim. Single-image deraining using an adaptive nonlocal means filter. In *2013 IEEE International Conference on Image Processing*, pages 914–917. IEEE, 2013.
- [14] L. Kratz and K. Nishino. Factorizing scene albedo and depth from a single foggy image. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1701–1708. IEEE, 2009.
- [15] 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*, pages 2736–2744, 2016.
- [16] Y. Luo, Y. Xu, and H. Ji. Removing rain from a single image via discriminative sparse coding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3397–3405, 2015.
- [17] S. G. Narasimhan and S. K. Nayar. Contrast restoration of weather degraded images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):713–724, June 2003.
- [18] K. Nishino, L. Kratz, and S. Lombardi. Bayesian defogging. *International journal of computer vision*, 98(3):263–278, 2012.
- [19] S. Ono, T. Miyata, and I. Yamada. Cartoon-texture image decomposition using blockwise low-rank texture characterization. *Image Processing, IEEE Transactions on*, 23(3):1128–1142, 2014.
- [20] C. Osendorfer, H. Soyer, and P. van der Smagt. *Neural Information Processing: 21st International Conference, ICONIP 2014, Kuching, Malaysia, November 3-6, 2014. Proceedings, Part III*, chapter Image Super-Resolution with Fast Approximate Convolutional Sparse Coding, pages 250–257. Springer International Publishing, Cham, 2014.
- [21] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, jan 2014.
- [22] S.-C. Pei and T.-Y. Lee. Nighttime haze removal using color transfer pre-processing and dark channel prior. In *2012 19th IEEE International Conference on Image Processing*, pages 957–960. IEEE, 2012.
- [23] H. Schaeffer and S. Osher. A low patch-rank interpretation of texture. *SIAM Journal on Imaging Sciences*, 6(1):226–262, 2013.
- [24] G. Steidl, J. Weickert, T. Brox, P. Mrázek, and M. Welk. On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and sides. *SIAM J. Numerical Analysis*, 42(2):686–713, 2004.
- [25] S.-H. Sun, S.-P. Fan, and Y.-C. F. Wang. Exploiting image structural similarity for single image rain removal. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4482–4486. IEEE, 2014.
- [26] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [27] B. Wohlberg. Efficient convolutional sparse coding. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7173–7177. IEEE, 2014.
- [28] B. Wohlberg. Efficient algorithms for convolutional sparse representations. *IEEE Transactions on Image Processing*, 25(1):301–315, Jan 2016.
- [29] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2528–2535. IEEE, 2010.
- [30] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *International Conference on Computer Vision*, pages 2018–2025, Nov 2011.
- [31] H. Zhang and V. M. Patel. Convolutional sparse coding-based image decomposition. In *British Machine Vision Conference*, 2016.

- [32] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng. Rain removal in video by combining temporal and chromatic properties. In *2006 IEEE International Conference on Multimedia and Expo*, pages 461–464. IEEE, 2006.
- [33] X. Zheng, Y. Liao, W. Guo, X. Fu, and X. Ding. Single-image-based rain and snow removal using multi-guided filter. In *International Conference on Neural Information Processing*, pages 258–265. Springer, 2013.