Removing rain from a single image via discriminative sparse coding*

Yu Luo[1,2], Yong Xu[1], Hui Ji[2]
[1]School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China
[2]Department of Mathematics, National University of Singapore, Singapore 119076
{l.y97@mail.scut.edu.cn, yxu@scut.edu.cn, matjh@nus.edu.sg}

Abstract

Visual distortions on images caused by bad weather conditions can have a negative impact on the performance of many outdoor vision systems. One often seen bad weather is rain which causes significant yet complex local intensity fluctuations in images. The paper aims at developing an effective algorithm to remove visual effects of rain from a single rain image, i.e. *separate the rain layer and the de-rained image layer from a rain image. Built upon a non-linear generative model of rain image, namely screen blend model, we propose a dictionary learning based algorithm for single image de-raining. The basic idea is to sparsely approximate the patches of two layers by very high discriminative codes over a learned dictionary with strong mutual exclusivity property. Such discriminative sparse codes lead to accurate separation of two layers from their non-linear composite. The experiments show that the proposed method outperforms the existing single image de-raining methods on tested rain images.*

1. Introduction

Most outdoor vision systems, *e.g*. surveillance and autonomous navigation, require accurate feature detection of images of outdoor scenes. Under bad weather, the content and color of images are often drastically altered or degraded [16]. Based on the size of constituent particles in atmosphere, the adverse weather can be classified into two categories [9]: steady bad weather caused by microscopic particles including fog, mist, haze, smoke [11]; and dynamic bad weather caused by large particles including rain, snow and hail. The impact of steady bad weather on images is relatively spatially consistent, and usually it leads to global loss of image contrast and color fidelity [12]. The resulting effect is that many image details are lost or hardly visible.

*Project supported by National Nature Science Foundations of China (61273255 and 61070091), Engineering & Technology Research Center of Guangdong Province for Big Data Analysis and Processing ([2013]1589-1-11) and Singapore MOE Research Grant R-146-000-178-112.

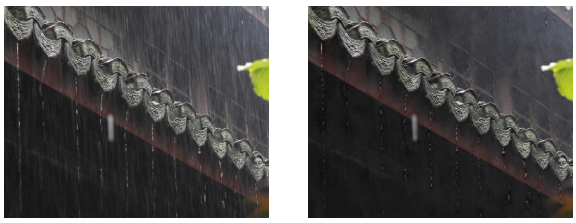


Figure 1. Left: rain image; and right: de-rained image

Under the dynamic bad weather, the large particles can be randomly distributed and cause complex intensity fluctuations in small image regions. As a result, these significant local intensity fluctuations will have noticeable negative impact on the reliability of image feature detection modules of vision systems. Therefore, for the reliability of outdoor vision systems, there is certainly such a need to develop effective methods to remove undesired visual effects caused by bad weather from images.

In the last few decades, there has been an abundant literature on recovering outdoor images related to bad weathers, *e.g*. image recovery for atmospherically blurred images [25], contrast enhancement of weather degraded images [16], and removing haze from images [12]. For recovering images taken in rainy weather, most of the work has been focusing on video or data with multiple frames [3, 10]. As visual systems often use images from different views for recognition tasks or the video might contains very large movements, single image based approach for rain removal has its necessities in practice. In this paper, we are interested in studying the image recovery problem for outdoor images taken in rainy weather, *i.e*., how to remove visual effects of rain from a single image; see Fig. 1 for an illustration.

Rain image recovery. There are two sub-problems in rain image recovery: 1) how to identify rain in images, and 2) how to remove visual effects caused by rain from images. Indeed both of them are problems difficult to solve.

- *Rain detection*. It is a challenging task to accurately detect rain streaks in a single image. The visual effects

of rain are quite complex [9], as rain contains a group of randomly distributed raindrops falling at high velocities. Visually, rain can be modeled as bright streaks with elongated elliptical shapes whose physical properties, *e.g*. brightness, spatial distributions, raindrop size and orientation, vary among different images.

- *Rain removal*. Removing visual effect of rain from a single image is also a difficult problem. It is known that the visual effect of rain in image can not be well modeled by a linear additive model of a rain layer and an image layer. These two layers are blended in a complex manner; see *e.g*. [9].

In the past, the study of rain image recovery has been focused on processing videos or image sequences; see *e.g*. [3, 4, 5, 6, 9, 10, 19, 20, 21, 28]. The redundant temporal information existing in videos or multiple consecutive frames provides rich information to identify rain drops. The work on processing single rain image has been scant in the existing literature. Built upon the concept of sparse coding, this paper aims at developing a variational approach for single rain image recovery.

Main idea. In this paper, we propose a variational model that simultaneously detects and removes rain streaks from an input image. Instead of using linear additive composite model adopted by many existing methods (*e.g*. [3, 14]), we propose to use a non-linear composite model called *screen blend model* for modeling rain images, one built-in feature in Photoshop. In screen blend model, a rain image, denoted by J, is composited by two layers, de-rained image layer I and rain layer R, as follows:

$$J = I + R - I * R, \tag{1}$$

where $'*'$ denotes point-wise multiplication operator. Then, de-raining a rain image is about how to separate the layer I and the layer R from the non-linear model (1).

In this paper, we propose a sparsity-based regularization strategy to solve (1) which assumes that (i) the local patches from both image layer I and rain layer R can be sparsely modeled in a learned dictionary D, (ii) the sparse codes generated from the learned dictionary should have sufficient discrimination between two layers. One approach is using two dictionaries with radically different patterns to separate two layers, *e.g*., Elad *et al*. [23] separates the texture region and cartoon region using both wavelet filter dictionary and local DCT dictionary. Using fixed dictionaries certainly is not a good choice in our case as they are not adaptive to input images. Motivated by the idea of discriminative K-SVD method (*e.g*. [13, 18, 27]) for sparse coding based image classification, we propose a discriminative sparse coding method to regularize the image de-raining process under over learned dictionary with mutual exclusivity property.

More specifically, let $\mathcal{P}$ denote the linear operator that assembles an array of vectorized image patches collected from an image, and let $\mathcal{B}$ denote the vector of the ℓ_2 norm of each row of an array, which represents the importance weight of each dictionary atom often used in feature selection. Then, we propose the following variational model for image de-raining:

$$\min_{I,R,D,C_I,C_R} \|\mathcal{P}I - DC_I\|_F^2 + \|\mathcal{P}R - DC_R\|_F^2 \tag{2}$$

subject to $J = I + R - I * R$ and

$$\begin{cases} \|C_I[:,j]\|_0 \leq T_1,\ \|C_R[:,j]\|_0 \leq T_2, & \text{for all } j; \\ |\mathcal{B}(C_I)^\top \mathcal{B}(C_R)| \leq \epsilon. \end{cases} \tag{3}$$

The last constraint listed in (3) ensures that the dictionary learned for sparse coding has the mutual exclusivity property such that the atoms used for synthesizing de-rained image patches will not be used for synthesizing rain patches and vice versa. Consequently, when using such a dictionary, the codes associated with the patches of image layer are sufficient separated from the codes associated with the patches of rain layer. In other words, sufficient discrimination between two layers is provided for separation.

Advantages of our approach. There have been a few methods proposed for single image de-raining; see *e.g*. [7, 14, 15]. These existing methods either take an ad-hoc approach that calls additional image processing modules as pre- or post-process. For example, the two-stage method [15] requires calling some kernel regression methods for detecting rain streaks which are often not stable as we observed in the experiments. The method proposed in [14] requires the bilateral filtering for preprocessing, as well as clustering algorithm for dictionary separation. In contrast, our approach provides a variational model, together with a simple iterative numerical method, and there is no any other image processing module involved. The experiments also show that the proposed method consistently outperforms the existing methods in terms of visual quality.

The remainder of this paper is organized as follows. Section 2 gives a brief review on existing video and single image de-raining methods. Section 3 is devoted to the detailed discussion on the proposed single image de-raining method. The results on both synthetic images and real images are provided in Section 4. Section 5 concludes the paper.

2. Related work

In this section, we give a brief review on the most related work on processing rain images.

2.1. Video de-raining

In the past, most of the work on rain image recovery takes videos or multiple frames as the input; see *e.g*.

[3, 4, 5, 6, 9, 10, 19, 20, 21, 28]. The redundant temporal information in videos allows more accurate detection of rain streaks, as well as high-quality recovery from the fusion of multiple frames. Garg and Nayar [9] presented a general analysis on the visual effects of various bad whether conditions, including rain, in an imaging system. A correlation model is proposed in [9] to capture the dynamics of rain. Once the rain region is detected, the intensity of rain pixel can be re-estimated by using the average of its non-rain temporal neighboring pixels. Zhang *et al*. [28] proposed two rain priors to detect the rain regions. One is the temporal prior which says rain cannot cover every pixel throughout the entire video, and the other is that the chromatic changes caused by raindrops are almost the same across three color channels $[R, G, B]$. Brewer and Liu [5] proposed another strategy for detecting rain streaks by checking the image gradient orientation information and by checking whether the region exhibits a short duration intensity spike. Based on the photometry properties and size information of rain streaks, Bossu *et al*. [4] derived some rules for selecting the potential rain streaks, and then a Gaussian Mixture model is used to separate the rain layer from the background. Barnum *et al*. [3] created a model of the visual effect of rain in frequency space, and analyzed the properties of rain layer in the transformed domain. In [26], with the analysis of the local spatial-temporal derivatives of raindrops, the partially or completely occluded image pixels by rain streaks are recovered by using some image completion technique. V Santhaseelan and VK Asari [21] utilized the phase congruency features to detect the rain streaks and estimated the intensity of rain pixels based on the minimization of registration error between frames.

2.2. Single image de-raining

Compared to video de-raining, less studies have been done on single image de-raining. The single image de-raining is generally harder than video de-raining in terms of visual quality, as less information is available for detecting and removing rain steaks from images. Recently, there have been a few approaches [14, 15, 17] proposed to tackle single image de-raining problem. Next we will give a brief discussion on these approaches. Kang *et al*. [14] proposed a bilateral filtering based approach for single image de-raining. In [14], the input rain image is first decomposed into two layers by bilateral filtering: one piece-wise smooth layer called "Layer A" and the other layer called "Layer B" dominated by small intensity oscillation. By taking Layer A as a fine estimation of de-rained image, the estimation of de-rained image is further refined by adding back the geometric part that remains in Layer B. The detection of such part is done via estimating the sparse approximation of Layer B under a non-rain dictionary, which is separated from the general dictionary learned from the whole layer B via K-means clustering. The method proposed in [14] works very well on images of simple structures that can be well approximated by the piece-wise model, but performs poorly on images of complex structures. One possible cause is that the bilateral filtering tends to yield piece-wise smooth results which will erase details that cannot be easily recovered later, and another possible cause is due to the weakness of K-means clustering to separate out the rain atoms and non-rain atoms.

Kim *et al*. [15] proposed a two-stage approach for image de-raining. The first stage of their approach is to detect rain streaks by a kernel regression method, and the second stage is to recover the rain regions by a non-local mean filtering which essentially uses the non-rain image patches to recover the rain patches. There exist two issues: one is that the kernel regression tends to miss a noticeable percentage of rain steaks, the other one is that the visual quality of the de-rained results by the non-local mean filtering still leaves plenty of room for improvement as the rain drops are treated as some additional noise which does not hold true for real images. A similar approach is proposed in [17]. The differences between these two two-stage approaches lie in the color space used, the detection strategy and the recovery of rain pixels using the information from non-rain pixels. Owing to the nature of the two-stage approach, the one by [17] suffers from the same issues as the one by [15], *e.g*. the inaccurate detection and unsatisfactory in-painting results.

A related application to the single image de-raining is image recovery for images taken through glass window in rainy weather [7]. Owing to the additional reflection layer of glass window, the physical model in [7] is not the same as the one discussed in this paper which considers the images taken outdoors. The method proposed in [7] is based on the technique of deep learning. In [7], a convolution neural network, a mapping between image patches corrupted by rain and clean image patches, is learned in [7] and then is used to predict de-rained image patches from rain image patches.

The approach proposed in this paper is a variational one that simultaneously detects and removes rain streaks via an iterative numerical scheme. Rain image is modeled by a non-linear composite model between two layers, rain layer and de-rained image layer which are estimated via the discriminative sparse coding. Unlike the methods proposed in [14, 15, 17], the proposed method does not rely on other image processing modules for pre- or post-processing, which avoids the likely vulnerability of these techniques when processing images of complex structures. The experiments also demonstrate the advantages of the proposed method over these existing methods.

3. Single image de-raining via sparse coding

In this paper, we first propose a new variational model for single image de-raining, and then present a greedy pur-

suit algorithm. Before starting the discussion, we give an introduction to the notations used in this section. Given a matrix Y, let $Y[:,j]$ denote its j-th column, and let $Y[i,j]$ denote the (i,j)-th element of Y, the Frobenius norm of Y is defined by $\|Y\|_F = (\sum_{i,j} y[i,j]^2)^{1/2}$. Given a vector y, let $y[j]$ denote its j-th entry, $\mathrm{diag}(y)$ denote the diagonal matrix whose diagonal entries defined by y, and let $\|y\|_0$ denote its ℓ_0 norm which counts the number of non-zero entries of y.

Screen blend model for modelling rain images. Many models have been proposed for rendering rain images. The most often seen ones in the literature are linear additive composite models. One popular additive composite model (see, *e.g.* [3, 14]) assumes that the rain image J is composited by two layers, de-rained image layer I and rain layer R, as follows:

$$J = I + R. \tag{4}$$

However, the rain images synthesized via (4) lack some characteristics often shown in real rain images, *e.g.*, the effects of internal reflections [8]. This motivates us to adopt a non-linear composite model, *screen blend model*. The screen blend model is used in Photoshop for rendering more realistic rain images. It composites two layers by inverting, multiplying and inverting two layers again:

$$J = 1 - (1 - I) * (1 - R) = I + R - I * R, \tag{5}$$

where '*' denotes the point-wise multiplication operator. The images rendered by the screen blend model usually look more realistic than those by the simple additive composite model. For example, for a dark background with brightness close to 0, the rain layer will dominate the appearance of the rain image. In contrast, when the background is quite bright with brightness close to 1, the rain image is dominated by the image layer and the rain layer is hardly visible. For a homogeneous gray region, using screen blend model is equivalent to using this gray value as opacity degree of blending a white region in a normal mode, which is similar to the transparency effect of rain drops.

Discriminative sparse coding for layer separation. Sparsity-based priors have been used in a wide range of image recovery tasks with impressive performance. In this paper, we also consider the sparsity prior for image layer and rain layer under a learned dictionary. More specifically, define Y_1 the collection of patches of image layer in vector form and Y_2 the collection of patches of rain layer, *i.e.*:

$$Y_1 := \mathcal{P}I; \quad Y_2 := \mathcal{P}R, \tag{6}$$

where $\mathcal{P}$ denotes the linear operator which maps the layer to the array of patches. Then, we assume that we can learn a dictionary D in an un-supervised setting such that both Y_1 and Y_2 can be sparsely approximated under the dictionary D:

$$Y_1 \approx DC_1; \quad Y_2 \approx DC_2,$$

where C_1 and C_2 are the sparse codes.

However, since only the composite of Y_1 and Y_2 is available, further discriminative term is needed to separate two layers I and R. Motivated by the discriminative K-SVD method [27] and label consistency K-SVD method [13] in sparse coding based classification. We assume that the patches of two layers can be classified into two different classes by their labels derived from the sparse codes. More specifically, let $\mathcal{B}$ denote the weight vector that calculates the ℓ_2 norm of each row of an array which is often seen in feature selection:

$$\mathcal{B}(C)[k] = \sum_j C[k,j]^2.$$

Then, the correlation of two weight vectors $\mathcal{B}(C_1)$ and $\mathcal{B}(C_2)$, defined by $|\mathcal{B}(C_1)^\top \mathcal{B}(C_2)|$, should be sufficiently small (*i.e.* smaller than a threshold ϵ_0) such that patches of two layers can be well classified into two different categories. In the extreme case with $\mathcal{B}(C_1)^\top \mathcal{B}(C_2) = 0$, the atoms used for generating the patches of rain layer will not be used for generating the patches of image layer, we call it mutual exclusivity property. In other words, we learn a dictionary with such a mutual exclusivity property to separate image patches and rain patches. Based on the discussion above, we proposed the following variational model for image de-raining:

$$\min_{I,R,D,C_1,C_2} \|\mathcal{P}I - DC_1\|_F^2 + \|\mathcal{P}R - DC_2\|_F^2 \tag{7}$$

subject to

$$\begin{cases} J = I + R - I * R; \\ 0 \le I \le 1; \quad 0 \le R \le 1; \\ \|C_1[:,j]\|_0 \le T_1,\ \|C_2[:,j]\|_0 \le T_2, \quad \text{for all } j; \\ |\mathcal{B}(C_1)^\top \mathcal{B}(C_2)| \le \epsilon_0. \end{cases} \tag{8}$$

where T_1 and T_2 are the sparsity constraints of each column of the sparse codes C_1 and C_2 respectively.

A greedy pursuit algorithm. The minimization problem (7) is a challenging non-convex optimization problem. One widely used technique for solving this type of problem is to alternately update five variables during each iteration. Such a multi-block alternating iteration is quite slow to converge to true solutions. In this section, we propose a greedy pursuit algorithm which empirically performs better than the plain implementation of alternating iteration scheme. The basic idea is as follows. In each iteration, firstly, we pursue the sparse approximation of the patches of image layer. Secondly, we pursue the sparse approximation of the patches of rain layer plus the residual left from the image layer, provided that the associated sparse codes are as much discriminative as possible. Lastly, we project the estimated patches to

fit the screen blend model and to learn the dictionary again, before starting the next iteration. See Algorithm 1 for the outline of the greedy pursuit algorithm for solving (7). All the optimization problems including the sparse coding and the dictionary updating in the algorithm can be solved by the routines used in the K-SVD method or by the proximal method presented in [1, 2].

It is empirically observed that the sequences I^k, R^k do converge in our experiments. It can be seen that all other constraints in (8) are explicitly enforced in Algorithm 1 except the penalty term on mutual exclusivity $|\mathcal{B}(C_1)^\top \mathcal{B}(C_2)|$. Indeed, Algorithm 1 heuristically finds an approximate solution to (7) in a greedy manner. To be more specific, there are two auxiliary variables ω and r in Algorithm 1. The variable ω is an indicator vector with dimensionality same as dictionary size, the atoms corresponding to the entries of ω with value 1 indicate that these atoms are used for rain layer, and for image layer otherwise. The variable r denotes the possible residue of rain layer remaining in image layer, which will be further suppressed by another round of the separation of rain layer and image layer. Suppose that the proposed algorithm does converge, we have that when $k \to \infty$, the term r^k will converge to zero, which implies that penalty term on mutual exclusivity $|\mathcal{B}(C_1)^\top \mathcal{B}(C_2)|$ converges to zero, that is to say, the last constraint in (8) can also hold true with sufficient iterations.

4. Experiments

In this section, Algorithm 1 is tested on both synthesized rain images and real rain images, and is compared against two existing single image de-raining methods, including Kang *et al.*'s method [14] and Kim *et al.*'s method [15]; see Section 2.2 for the details of these two methods. The convolution neural network based method [7] is developed to deal with those images taken through glass window. A direct application of this method does not work on the problem addressed in this paper, as we observed in the experiments. Thus, it is not included in the experimental evaluation. Kang *et al.*'s method is chosen as the source code is available [1]. Kim *et al.*'s method is chosen since we can well replicate the results provided in their paper. In this paper, the results of Kang *et al.*'s method are generated using the author's implementation with suggested parameter setting. And the results of Kim *et al.*'s method are directly quoted from the paper if available, and are generated from our implementation with rigorous parameter tune-up otherwise.

4.1. Experimental setting

Algorithm 1 is implemented in MATLAB R2014a and the experiments are carried in a windows PC with an INTEL CPU (2.4GHZ) and 16G memory.

[1] http://www.ee.nthu.edu.tw/cwlin/Rain_Removal/Rain_Removal.htm

Input : rain image J
Output: de-rained image I^K and rain layer R^K
Initial : dictionary D^0, sparse codes C_1^0, C_2^0, two patch arrays Y_1^0, Y_2^0, exclusive indicator w^0
for $k = 0, 1, \ldots, K$, **do**

(1) *sparse coding*:

$$C_1^k := \text{argmin}_C \|Y_1^{k-1} - D^k C\|_F^2 \quad \text{subject to } \|C[:,j]\|_0 \leq T_1;$$

$$r^k := D^k \text{diag}(\omega^{k-1}) C_1^k;$$

$$C_2^k := \text{argmin}_C \|Y_2^{k-1} + r^k - D^k C\|_F^2 \quad \text{subject to } \|C[:,j]\|_0 \leq T_2.$$

(2) *updating the patches of rain layers*:

$$Y_2^k := D^k C_2^k.$$

(3)*updating both layers*:

$$R^k \Leftarrow Y_2^k \quad \text{(clipped to } [0,1]);$$

$$I^k := (J - R^k)./(1 - R^k). \tag{9}$$

(4) *sampling patches:*

$$Y_1^k := \mathcal{P} I^k; Y_2^k := \mathcal{P} R^k;$$

$$\omega^k[j] = \begin{cases} 1 & \text{if } \mathcal{B}(C_2^k)[j] \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

$$w^k = w^{k-1} .* w^k.$$

(5) updating dictionary:

$$D^k := \text{argmin}_D \|Y_1^k - D C_1^k\|_F^2 + \|Y_2^k - D C_2^k\|_F^2.$$

subject to $\|D_j\|_2 = 1, \quad$ for all j.

end

Algorithm 1: Outline of the greedy pursuit algorithm for image de-raining

Parameters setting. Through out all the experiments, the parameters are set to be the same, two bounds of sparsity degree, T_1 and T_2, are set to be 5 and 8 respectively. The patch size of both image layer and rain layer is set to be 16×16 pixels. The number of atoms of the dictionary is set to be 5×128. The iteration number K is set to be 5.

Initialization. The initialization is done as follows. For the atoms of dictionary D, we generate two subsets: one is the set that accounts for image patches and the other is for rain patches. The subset of dictionary atoms corresponding to image patches is trained from the un-processed rain image patches. The subset of dictionary atoms correspond-

ing to rain patches are constructed via the following steps. Firstly, the atoms are initialized using Gaussian kernels with different randomly chosen center in the matrix of size 16×16. The size and s.t.d. of Gaussian kernel is 3×3 and 1. Secondly, these initial atoms are smoothed by a linear motion blur kernel of length 24 pixels, and the orientation of the motion kernel is the dominate gradient orientation of original rain image, which is estimated via checking the largest entry of the histogram of image gradient orientations with width 5°. The size ratio of two subsets is $1:4$. The indicator vector w is initialized according to the initial atoms, 1 for rain atoms and 0 for image atoms. In our implementation, the initial dictionary for image patch and the initial sparse codes are generated by the K-SVD method and the OMP method respectively, and during the algorithm iteration, the dictionary and sparse approximation are updated by the proximal method presented in [1].

4.2. Experimental evaluation

Time consumption. For a color image of pixel size 256×256, the total running time in seconds of Kang *et al.*'s method [14], Kim *et al.*'s method [15] and Algorithm 1 are 358, 252, 140 respectively. Algorithm 1 is faster.

Synthetic Data. To quantitatively evaluate the performance of the proposed method, the proposed method is tested on synthesized rain images. The rain images for testing are synthesized by adding rainy effect on 200+ outdoor images randomly selected from the UCID dataset [22]; see Fig. 3 for some samples. The rainy effect is generated by two existing composite models: one is additive composite model (4) and the other is screen blend model (5). Following the process described in the document[2], the rain images are synthesized with rain streaks whose orientations varying from 70° to 110°. Thus, we obtain two data sets for performance evaluation with one corresponding to additive composite model and one corresponding to screen blend model. The quality of the results are measured by peak signal to noise ratio (PSNR) and structural similarity (SSIM) [24].

It is noted that Algorithm 1 can also be used under model (4) by replacing (9) in Algorithm 1 with the following step:

$$I^k := J - R^k. \tag{10}$$

Table 2 lists the quality metric values of the de-rained results by three methods on the dataset synthesized by the additive composite model (4). It can be seen that using the same model, the proposed method noticeably outperforms the existing two methods.

For the dataset synthesized by the screen blend model (5), see Table 3 for the results of Algorithm 1.B. The comparison to other methods are not included, as they do not use such a model for rain images. The samples with rainy effect added using both screen blend model and additive model are shown in Fig. 2, as well as the results obtained from Algorithm 1.A and Algorithm 1.B. More results can be found in the supplementary material.

[2] http://www.photoshopessentials.com/photo-effects/rain/

Table 1. Two versions of Algorithm 1 for two rain image models

Version	Image model	Algorithm
Algorithm 1.A	linear composite model (4)	Algorithm 1 w/ (10)
Algorithm 1.B	screen blend model (5)	Algorithm 1 w/ (9)

Table 2. Results of synthetic images using addtive model (4)

Metric	Clear	Rainy	Kang[14]	Kim[15]	Algorithm 1.A
PSNR	-	22.02	22.23	22.33	25.79
SSIM	1	0.81	0.82	0.85	0.87

Table 3. Results of synthetic images using screen blend model (5)

Metric	Clear	Rainy	Algorithm 1.B
PSNR	–	25.71	28.74
SSIM	1	0.88	0.92

Figure 2. The first row shows the rain image synthesized using additive model (4) and the result from Algorithm 1.A; the second row shows the rain image synthesized using screen blend model (5) and the result of Algorithm 1.B.

Real data. The proposed method is also evaluated on real images. See Fig. 4–7 for the visual illustration of four real images. More results can be found in the supplementary material. The rain image shown in Fig. 4 (a) is from [15].

It can be seen that Algorithm 1.B outperforms the other two existing methods in terms of both the effectiveness of removing rain streaks and the visual quality of recovered images. For example, Kang *et al.*'s algorithm adds some artifacts on the face as shown in Fig. 4 (b), and fails to remove many rain streaks. Kim *et al.*'s results have block artifacts caused by the nonlocal mean filtering. In contrast, Algorithm 1.B is able to remove most of the rain streaks and meanwhile it produces less artifacts on the recovered image. The same is observed in the results shown in Fig. 5. Our results are noticeably better than the other two. As shown in Fig. 5 (b), there are many rain streaks remaining and many blurry artifacts introduced in the result obtained from Kang *et al.*'s method. Kim *et al.*'s method also fails to remove

Figure 3. Some samples from the UCID dataset

(a) Rain image (b) Kang *et al*. [14] (c) Kim *et al*. [15] (d) Algorithm 1.A (e) Algorithm 1.B

Figure 4. Rain image "obama" and the results . The first row shows the full image and the second row shows one zoomed-in region.

many rain streaks due to the inaccuracy of rain detection, and the detected region is in-painted with poor quality. For Fig. 6, one can see that our result removes most of the rain streaks and keeps most textural details of red leaves. Contrarily, Kang *et al*.'s method and Kim *et al*.'s method neither effectively remove rain streaks nor keep the textural details of the leaves. For Fig. 7, it can be seen that our method has the sharpest appearance of the trees and the lamp. As a comparison, the result from Kang *et al*.'s method is more blurry. The result of Kim *et al*.'s method suffers from the false detection of rain streaks, as the lamp with a nearly vertical orientation is wrongly removed.

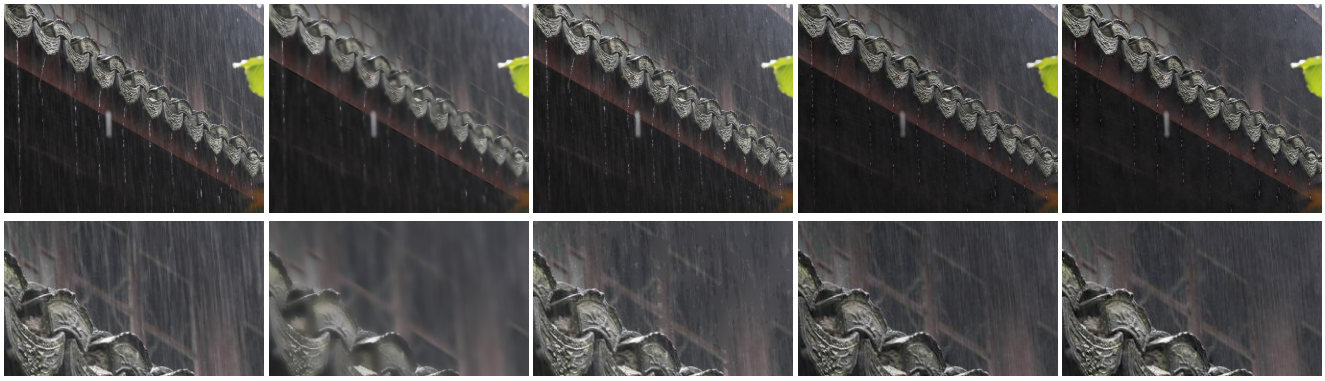
Also, Algorithm 1.A outperforms the other two existing methods in all images. This implies the advantage of the proposed discriminative sparse coding approach over the existing ones under the same rain image model. The additional benefit of introducing screen blend model is also justified by comparing the results from Algorithm 1.A with those from Algorithm 1.B. Generally speaking, these two algorithms show the comparable performance in terms of removing rain streaks, however, the linear additive model will generate artifacts in the bright region, see the left top corner of Fig. 6, or lower the luminance of the whole image, *e.g*. Fig. 7. This luminance change can be seen more apparently in the white leaves of Fig. 6. These results show that the screen blend model is a better model than the linear composite model when processing real rain images.

5. Discussions and Conclusions

In this paper, we develop a discriminative sparse coding based approach for single image de-raining. Different from the existing approaches, the proposed method considers a non-linear screen blend model for modeling rain images. By learning a dictionary with mutual exclusivity, the derained image layer and the rain layer can be accurately separated using sparse coding with high discriminability. The experiments shows that the proposed method clearly outperforms the existing methods on both synthesized and real images. Nevertheless, the results by the proposed method are not perfect. There are some artifacts around the regions containing rain streaks, which is due to the ambiguities between the low-pass frequency components of image layer and the low-pass frequency components of rain layer. Such ambiguities in low-pass channels cannot be completely resolved in the proposed method.

The proposed method also has some other limitations. One is that the proposed method might not work well when the input image has many structures similar as the raindrops, and the other is that the proposed one is not applicable to rain images with magnified raindrops such as water drops stick to the glasses. In future, we will work on establishing an image dataset for better evaluating image de-raining algorithms and refine the proposed method to overcome the limitations mentioned above. In addition, we will also investigate the applicability of the proposed discriminative sparse coding based method to other source separation problems in computer vision.

Acknowledgement. The authors would like to thank the reviewers and area chair for their very helpful comments and suggestions that greatly improve the quality of the manuscript.



(a) Rain image (b) Kang *et al*. [14] (c) Kim *et al*. [15] (d) Algorithm 1.A (e) Algorithm 1.B

Figure 5. Rain image "eave" and the results . The first row shows the full image and the second row shows one zoomed-in region.

(a) Rain image (b) Kang *et al*. [14] (c) Kim *et al*. [15] (d) Algorithm 1.A (e) Algorithm 1.B

Figure 6. Rain image "bush" and the results . The first row shows the full image and the second row shows one zoomed-in region.

(a) Rain image (b) Kang *et al*. [14] (c) Kim *et al*. [15] (d) Algorithm 1.A (e) Algorithm 1.B

Figure 7. Rain image "lamp" and the results . The first row shows the full image and the second row shows one zoomed-in region.

References

[1] C. Bao, H. Ji, Y. Quan, and Z. Shen. L0 norm based dictionary learning by proximal methods with global convergence. In *CVPR*, pages 3858–3865. IEEE, 2014. 5, 6

[2] C. Bao, Y. Quan, and H. Ji. A convergent incoherent dictionary learning algorithm for sparse coding. In *ECCV*, pages 302–316. Springer, 2014. 5

[3] P. C. Barnum, S. Narasimhan, and T. Kanade. Analysis of rain and snow in frequency space. *IJCV*, 86(2-3):256–274, 2010. 1, 2, 3, 4

[4] J. Bossu, N. Hautière, and J.-P. Tarel. Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *IJCV*, 93(3):348–367, 2011. 2, 3

[5] N. Brewer and N. Liu. Using the shape characteristics of rain to identify and remove rain from video. In *Struct. Syn., Stat. Patt. Recog.*, pages 451–458. Springer, 2008. 2, 3

[6] Y.-L. Chen and C.-T. Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *ICCV*, pages 1968–1975, 2013. 2, 3

[7] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *ICCV*, pages 633–640, 2013. 2, 3, 5

[8] K. Garg and S. K. Nayar. Photometric model of a rain drop. Technical report, Columbia University, 2003. 4

[9] K. Garg and S. K. Nayar. Detection and removal of rain from videos. In *CVPR*, volume 1, 2004. 1, 2, 3

[10] K. Garg and S. K. Nayar. Vision and rain. *IJCV*, 75(1):3–27, 2007. 1, 2, 3

[11] P. O. H. Tian, W. Li and L. Wang. Single image smoke detection. In *ACCV*, 2014. 1

[12] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. In *CVPR*, volume 33, pages 2341–2353, 2011. 1

[13] Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In *CVPR*, pages 1697–1704, 2011. 2, 4

[14] L.-W. Kang, C.-W. Lin, and Y.-H. Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Trans. Image Process.*, 21(4):1742–1755, 2012. 2, 3, 4, 5, 6, 7, 8

[15] J.-H. Kim, C. Lee, J.-Y. Sim, and C.-S. Kim. Single-image deraining using an adaptive nonlocal means filter. In *ICIP*, pages 914–917, 2013. 2, 3, 5, 6, 7, 8

[16] S. G. Narasimhan and S. K. Nayar. Contrast restoration of weather degraded images. *IEEE Trans. on PAMI*, 25(6):713–724, 2003. 1

[17] S.-C. Pei, Y.-T. Tsai, and C.-Y. Lee. Removing rain and snow in a single image using saturation and visibility features. In *ICME Workshops*, pages 1–6. IEEE, 2014. 3

[18] G.-J. Peng and W.-L. Hwang. Reweighted and adaptive morphology separation. *SIAM Journal on Imaging Sciences*, 7(4):2078–2104, 2014. 2

[19] M. Roser and A. Geiger. Video-based raindrop detection for improved image registration. In *ICCV Workshops*, pages 570–577, 2009. 2, 3

[20] M. Roser, J. Kurz, and A. Geiger. Realistic modeling of water droplets for monocular adherent raindrop recognition using bezier curves. In *ACCV Workshops*, pages 235–244, 2011. 2, 3

[21] V. Santhaseelan and V. K. Asari. Utilizing local phase information to remove rain from video. *IJCV*, pages 1–19, 2014. 2, 3

[22] G. Schaefer and M. Stich. Ucid: an uncompressed color image database. In *Electronic Imaging 2004*, pages 472–480. International Society for Optics and Photonics, 2003. 6

[23] J.-L. Starck, M. Elad, and D. L. Donoho. Image decomposition via the combination of sparse representations and a variational approach. *Image Processing, IEEE Transactions on*, 14(10):1570–1582, 2005. 2

[24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE trans. Image Process.*, 13(4):600–612, 2004. 6

[25] Y. Yitzhaky, I. Dror, and N. Kopeika. Restoration of altmospherically blurred images according to weather-predicted atmospheric modulation transfer function. *Optical Eng.*, 36, 1998. 1

[26] S. You, R. T. Tan, R. Kawakami, and K. Ikeuchi. Adherent raindrop detection and removal in video. In *CVPR*, pages 1035–1042, 2013. 3

[27] Q. Zhang and B. Li. Discriminative K-SVD for dictionary learning in face recognition. In *CVPR*, pages 2691–2698, 2010. 2, 4

[28] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng. Rain removal in video by combining temporal and chromatic properties. In *ICME*, pages 461–464, 2006. 2, 3

Supplementary material for the Paper "Removing rain from a single image via discriminative sparse coding"

Yu Luo[1,2], Yong Xu[1], Hui Ji[2]
[1]School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China
[2]Department of Mathematics, National University of Singapore, Singapore 119076
{l.y97@mail.scut.edu.cn, yxu@scut.edu.cn, matjh@nus.edu.sg}

1. Experimental evaluation

In addition to the results shown in the main paper, more experiments are shown in this section. First, three more examples are shown in Fig. 1 and Fig. 2 for the synthesized rain images. Fig. 1 shows the rain images synthesized by the screen blend model and the de-rained results of Algorithm 1.B. For rain images synthesized by the linear additive model, the comparison of Algorithm 1.A to Kang *et al.*'s method [1] and Kim *et al.*'s method [2] is shown in Fig. 2. The synthesis process is the same as that in the main paper. It can be seen that the proposed method can effectively remove the rain streaks from the image without damaging the original image content.

More experiments on real rainy images are included in this supplementary material; see Fig. 3–Fig. 6. It is noted that the results from Kang *et al.* s method [1] are generated using the author's implementation with suggested parameter setting. And the results from Kim *et al.*'s method [2] are either directly quoted from the paper if available (*e.g.* Fig. 3), or are generated from our own implementation with rigorous parameter tune-up.

Similar to the results demonstrated in the main paper, in these experiments, the proposed method (Algorithm 1.A and Algorithm 1.B) outperforms the other two methods in terms of both the effectiveness of removing rain streaks and the visual quality of recovered images. For example, Kang *et al.*'s method fails to remove many rain streaks from the image as shown in Fig. 3 (b). Kim *et al.*'s method removes most of the rain streaks, but it also removes the details of the bike due to the false detection. In contrast, the proposed method is able to remove most of the rain streaks while keeping most of the details. For Fig. 4, one can see that Kang *et al.*'s method removes most rain streaks, but the textures are not well preserved, *e.g.*, the leaves are blurred. Kim *et al.*'s method produces a better result than Kang *et al.*'s method in terms of preserving textual details. However, its visual quality is inferior to ours. The same blurring artifacts of Kang *et al.*'s results can also be observed in Fig. 5(b). For Fig. 6, it can be seen that Kang *et al.*'s method removes many details while erasing most of the rain streaks. Kim *et al.*'s method performs slightly better than Kang *et al.*'s method, but many details in the result are still missing, *e.g.*, the digits of the license plate is hardly recognizable. As a comparison, the result from the proposed method is the only one that not only removes most rain streaks, but also keeps most of image details, *e.g.*, the digits of the license plate remain recognizable in our result.

References

[1] 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. 1, 2, 3

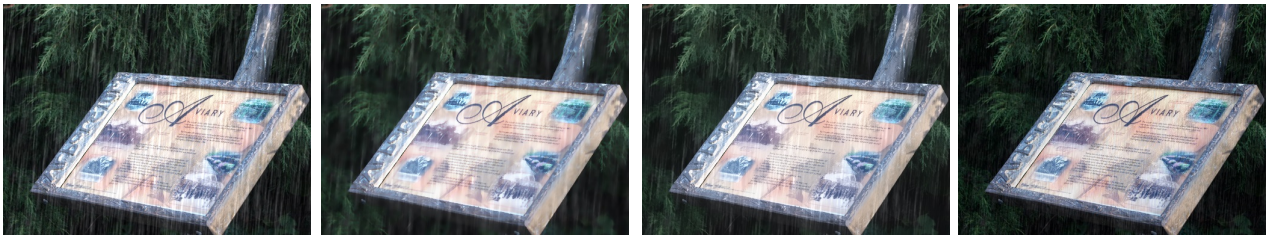
[2] J.-H. Kim, C. Lee, J.-Y. Sim, and C.-S. Kim. Single-image deraining using an adaptive nonlocal means filter. In *ICIP*, pages 914–917, 2013. 1, 2, 3

Original image Rain image (PSNR:24.2) Algorithm 1.B (PSNR:30.26)

Original image Rain image (PSNR:26.9) Algorithm 1.B (PSNR:30.3)

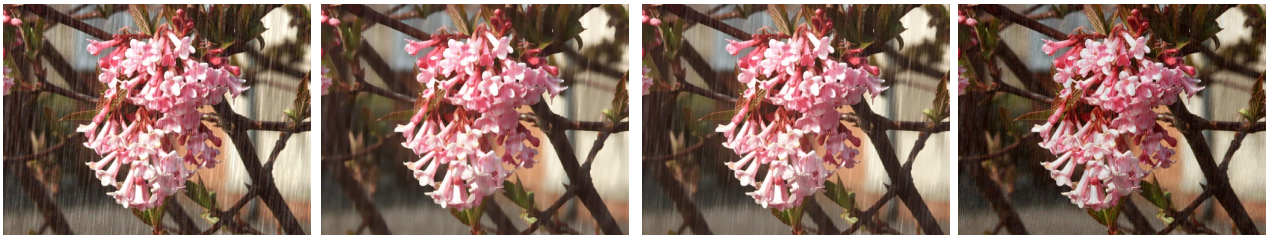
Original image Rain image (PSNR:24.8) Algorithm 1.B (PSNR:30.4)

Figure 1. Visual illustration of removing rainy effects generated by the screen blend model .



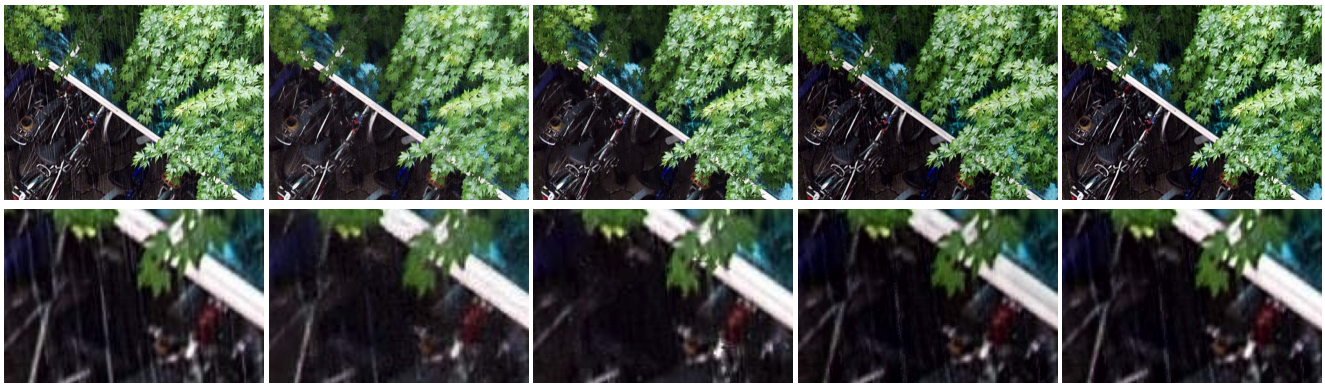
Rain image (PSNR:21.5) Kang *et al.*'s method[1] (PSNR:22.0) Kim *et al.*'s method [2] (PSNR:22.1) Algorithm 1.A (PSNR:28.9)

Rain image (PSNR:22.2) Kang *et al.*'s method[1] (PSNR:22.2) Kim *et al.*'s method [2] (PSNR:22.5) Algorithm 1.A (PSNR:27.1)



Rain image (PSNR:21.8) Kang *et al.*'s method[1] (PSNR:22.2) Kim *et al.*'s method [2] (PSNR:22.1) Algorithm 1.A (PSNR:27.7)

Figure 2. Visual comparison of removing rainy effects generated by the linear additive model.



(a) Rain image (b) Kang *et al*. [1] (c) Kim *et al*. [2] (d) Algorithm 1.A (e) Algorithm 1.B

Figure 3. Visual comparison of the rain image "Bicycle" and the de-rained results. The first row shows the full image and the second row shows one zoomed-in region.

(a) Rain image (b) Kang *et al*. [1] (c) Kim *et al*. [2] (d) Algorithm 1.A (e) Algorithm 1.B

Figure 4. Visual comparison of the rain image "tree" and the de-rained results. The first row shows the full image and the second row shows one zoomed-in region.

(a) Rain image (b) Kang *et al*. [1] (c) Kim *et al*. [2] (d) Algorithm 1.A (e) Algorithm 1.B

Figure 5. Visual comparison of the rain image "bridge" and the de-rained results.

(a) Rain image (b) Kang *et al*. [1] (c) Kim *et al*. [2] (d) Algorithm 1.A (e) Algorithm 1.B

Figure 6. Visual comparison of the rain image "carpark" and the de-rained results. The first row shows the full image and the second row shows one zoomed-in region.