Rain Streak Removal Using Layer Priors

Yu Li[1] Robby T. Tan[2,3] Xiaojie Guo[4] Jiangbo Lu[1] Michael S. Brown[3]

[1]Advanced Digital Sciences Center, Singapore [2]Yale-NUS College, Singapore
[3]National University of Singapore, Singapore
[4]State Key Lab of Information Security, IIE, CAS, China

Abstract

This paper addresses the problem of rain streak removal from a single image. Rain streaks impair visibility of an image and introduce undesirable interference that can severely affect the performance of computer vision algorithms. Rain streak removal can be formulated as a layer decomposition problem, with a rain streak layer superimposed on a background layer containing the true scene content. Existing decomposition methods that address this problem employ either dictionary learning methods or impose a low rank structure on the appearance of the rain streaks. While these methods can improve the overall visibility, they tend to leave too many rain streaks in the background image or over-smooth the background image. In this paper, we propose an effective method that uses simple patch-based priors for both the background and rain layers. These priors are based on Gaussian mixture models and can accommodate multiple orientations and scales of the rain streaks. This simple approach removes rain streaks better than the existing methods qualitatively and quantitatively. We overview our method and demonstrate its effectiveness over prior work on a number of examples.

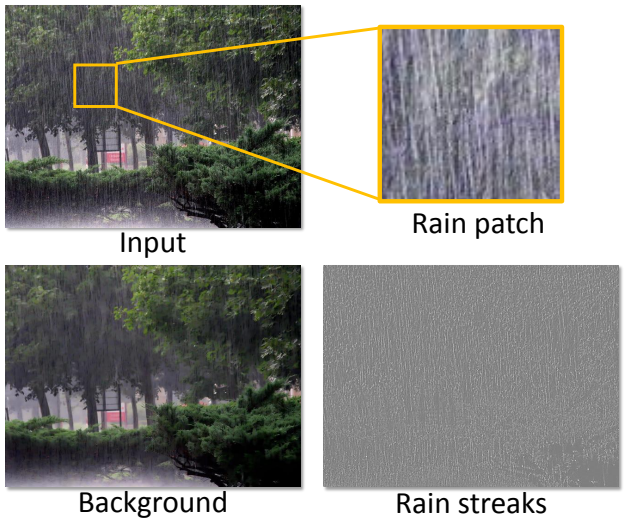

Figure 1: **Upper Row:** An example rain image and a zoomed-in patch that mainly contains annoying effect of rain streaks. Our method learns a rain streak layer prior on this region and used it for layer separation. **Lower Row:** The background and rain layers recovered by our proposed method, respectively.

1. Introduction

Most computer vision algorithms assume that the input image is of scene content that are clear and visible. However, for outdoor images, undesirable interference from rainy weather is often inevitable. Rain introduces several different types of visibility degradation. Raindrops that fall and flow on a camera lens or a windscreen can obstruct, deform and/or blur the imagery of the background scenes. Distant rain streaks accumulated throughout the scene reduce the visibility in a manner similar to fog, namely by scattering light out and into the line of sight, creating a veiling phenomenon. Nearby rain streaks, where the individual rain streaks are visible can also significantly degrade visibility due to their specular highlights, scattering, and blurring effect. Figure 1 shows an example of visibility degradation due to rain streaks and our attempt to remove them.

Mathematically, the observed rain image $\mathbf{O} \in \mathbb{R}^{M \times N}$ can be modeled as a linear superimposition [2, 11] of the desired background layer $\mathbf{B} \in \mathbb{R}^{M \times N}$ and the rain streak layer $\mathbf{R} \in \mathbb{R}^{M \times N}$, expressed as $\mathbf{O} = \mathbf{B} + \mathbf{R}$. The goal of rain streak removal is to decompose the rain-free background $\mathbf{B}$ and the rain streak layer $\mathbf{R}$ from an input image $\mathbf{O}$, and hence enhance the visibility of the image. This layer separation problem is an ill-posed one as the number of unknowns to be recovered is twice as many as that of the input. One common strategy is to use multiple images, or a video sequence, to mitigate the difficulty of the background scene recovery given the rich temporal information. In this paper,

however, we focus on the problem of rain streak removal given a single image only.

Rain streak removal for single image input is important for a couple of reasons. First, in some situations, we have only single image of a rainy scene (*e.g*., archived images, images available on the Internet, images taken by still cameras, *etc*.). Second, for dynamic scenes such as when the camera moves and/or part of the scene is dynamic, temporal information might not be reliable. As a result, being able to remove rain streaks for each frame will benefit the overall final result, since most video based methods assume a static background.

To make the problem well-posed and tractable, we enforce layer priors on both the background and rain components. More specifically, the idea of using patch-based priors is inspired by Zoran and Weiss [29] who used Gaussian mixture models (GMMs) to model image patches. This approach is simpler to compute than the existing prior models, such as FoE [19] or Weiss-Freeman's priors [25]. Recent works from Shih *et al*. [21] shows the superiority of using GMMs as priors in solving the reflection removal problems [14]. In our rain removal approach, to model the background patch priors, we use a GMM trained on patches from natural images. An additional gradient sparsity constraint is imposed to further regularize the background. As for the rain layer, we do the same, namely by gathering image patches of rain streaks that are located on textureless background. Unlike existing methods in single-image rain streak removal ([2, 11]), our method is easy to implement and generates considerably better results qualitatively and quantitatively. Moreover, our method is not limited by the rain streak orientations and scales, thus making it more practical. To our knowledge, this is the first method to use GMM patch priors for the purpose of rain streak removal.

The remainder of this paper is organized as follows. Section 2 discusses the related methods that deal with rain, including video-based rain streak removal and single-image based rain streak removal. Section 3 details our method, including the problem formulation and optimization. Section 4 shows the results and analyzes them in comparison with the results of other methods. Finally, the paper is concluded in Section 5.

2. Related work

There are a number of methods proposed to improve the visibility of images captured with rain streak interference. These fall into two categories: multiple image/video-based and single image methods.

Video Based Methods Early methods to remove rain streaks include work by Garg and Nayar [4, 7], which introduces a rain streak detection and removal method from a video sequence. The detection is based on two constraints: first, because rain streaks are dynamic, their changes in intensity within a few frames are relatively high. Second, since other objects are also possibly dynamic, rain streaks can be differentiated from these objects by verifying whether the intensity changes along the streak are photometrically linearly related to the background intensity. This second constraint will reduce the false alarms introduced by the first constraint. Having detected the rain streaks, the method removes the streaks by taking the average intensity of the pixels taken from the previous and subsequent frames.

Garg and Nayar [5] propose another method that exploits the ability to control a video camera's operational parameters when capturing a rainy scene. To this end, they show that rain visibility in images relies significantly on the exposure time and depth of field of the camera. Thus adjusting these parameters while taking the video will allow us to reduce the appearance of the rain streaks. Zhang *et al.* [27] added an additional constraint called the chromaticity constraint. They found the intensity changes in the RGB color channels are the same for pixels representing rain streaks. More recently, Bossu *et al.* [1] proposed a rain detection algorithm based on the histogram of streak orientations. The main idea is to fit a Gaussian distribution on rain streak histograms, such that they can be separated from noise generated by other dynamic objects. The method uses background subtraction to obtain the histograms. For a more complete review on the existing video-based rain streak removal, please refer to [23].

Single Image Methods For single image rain streak removal, the work by Kang *et al.* [11] proposed a method that decomposed an input image into its low frequency component (the structure layer) and a high frequency component (the texture layer). The high frequency component contains rain streaks and edges of background objects. This method attempts to separate the rain streak frequencies from the high frequency layer via sparse coding based dictionary learning with HoG features. The output is obtained by combining back the low frequency and processed high frequency layers. While the decomposition idea is elegant, the overall framework in [11] is complex and the results are not optimal. Results produced by this method tend to blur the background. These problems remain in the follow-up work for this method *e.g*. [10, 22]. More recently, Chen and Hsu [2] introduced a single objective function that could be used to decompose the background and rain streak layers. They formulated a cost function with three terms: a likelihood, a smoothed background layer, and a low rank rain streak layer. Although the idea of posing the problem into an objective function is attractive, the constraints are not sufficiently strong. In our experiments, we still observe a large amount of rain streaks remain in the output. Kim *et al*.

[12] detect rain streaks by assuming the elliptical shape and the vertical orientation of the rain, and remove the detected streaks using nonlocal mean filtering. This idea works for some cases of rain streaks, but unfortunately detecting individual streaks is challenging, because they could possibly have different orientations, scales, and densities. The recent work of [16] used discriminative sparse coding to remove rain streaks in a single image, but its effectiveness is still weak as one can see some residual thin structures at the rain streak locations in the output background images.

Aside from dealing with rain streaks, several methods have been proposed to address artifacts that arise when raindrops adhere to the camera lens or a windscreen in front of the camera (e.g. [3, 13, 18, 26]). The problems specific to adherent raindrops, however, are notably different from the interference caused by rain streaks. In particular, static adherent raindrops, which is the problem most existing methods target, are generally less dense than rain streaks. In addition, their size in an image is generally much larger than rain streaks, and tends to completely occlude parts of the background scene.

3. Our Method

3.1. Problem Formulation

The observed rain image $\mathbf{O} \in \mathbb{R}^{M\times N}$ can be modeled as a linear superimposition of the desired background layer $\mathbf{B} \in \mathbb{R}^{M\times N}$ and the rain streak layer $\mathbf{R} \in \mathbb{R}^{M\times N}$, such that:

$$\mathbf{O} = \mathbf{B} + \mathbf{R}. \tag{1}$$

The goal of rain streak removal is to decompose the rain-free background $\mathbf{B}$ and the rain streak layer $\mathbf{R}$ from a given input image $\mathbf{O}$. As previously stated, this problem is ill-posed. To resolve it, we propose to maximize the joint probability of the background layer and the rain layer using the MAP (maximum a posteriori): *i.e.* maximize $p(\mathbf{B},\mathbf{R}|\mathbf{O}) \propto p(\mathbf{O}|\mathbf{B},\mathbf{R}) \cdot p(\mathbf{B}) \cdot p(\mathbf{R})$ with the assumption that the two layers $\mathbf{B}$ and $\mathbf{R}$ are independent. Equivalently, with slight algebraic manipulation on the negative log function, we obtain the following energy minimization problem:

$$\begin{aligned} \min_{\mathbf{B},\mathbf{R}} \quad & \|\mathbf{O}-\mathbf{B}-\mathbf{R}\|_F^2 + \Phi(\mathbf{B}) + \Psi(\mathbf{R}) \\ \text{s.t.} \quad & \forall i \;\; 0 \le \mathbf{B}_i, \mathbf{R}_i \le \mathbf{O}_i, \end{aligned} \tag{2}$$

where $\|\cdot\|_F$ represents the Frobenius norm and i is the pixel index. The first term $\|\mathbf{O}-\mathbf{B}-\mathbf{R}\|_F^2$ helps maintain the fidelity between the observed and the recovered signals, while $\Phi(\mathbf{B})$ and $\Psi(\mathbf{R})$ designate the priors that will be respectively imposed on $\mathbf{B}$ and $\mathbf{R}$ to regularize the inference. The inequality constraint ensures that the desired $\mathbf{B}$ and $\mathbf{R}$ are positive images. More importantly, this inequality constraint plays a critical role in estimating reliable solutions as it regularizes the DC component of the recovered layers as verified in recent works by [15, 21].

Focusing our discussion on the priors, we first define the priors of the background layer as:

$$\Phi(\mathbf{B}) := -\gamma \sum_i \log \mathcal{G}_{\mathbf{B}}(\mathcal{P}(\mathbf{B}_i)) + \alpha\|\nabla\mathbf{B}\|_1, \tag{3}$$

where $\gamma(=0.01)$ and $\alpha(=0.05)$ are two non-negative coefficients balancing the corresponding terms. The function $\mathcal{P}(\mathbf{B}_i)$ is to extract the $n\times n$ (pre-defined size) patch around pixel $\mathbf{B}_i$ and reshape it into a vector of length n^2 with the DC component removed. The term $\mathcal{G}(\mathbf{x})$ stands for the GMM of $\mathbf{x}$, *i.e.* $\mathcal{G}(\mathbf{x}) := \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \mathbf{\Sigma}_k)$, where K is the total number of Gaussian components, π_k is the component weight such that $\sum_{k=1}^{K}\pi_k = 1$, while μ_k and $\mathbf{\Sigma}_k$ are the mean and covariance corresponding to the kth component, respectively. As the function $\mathcal{P}(\cdot)$ has removed the mean of every patch, $\mu_k = \mathbf{0}$ for all k. The benefit of this patch regularizer based on GMM has been demonstrated in [29]. In addition, it has been widely recognized that natural images are largely piecewise smooth and their gradient fields are typically sparse. Therefore we employ $\|\nabla\mathbf{B}\|_1$ to achieve such a functional, where ∇ denotes the gradient operator and $\|\cdot\|_1$ is the ℓ^1 norm.

As for the priors of the rain layer, we write it in the following form:

$$\Psi(\mathbf{R}) := -\gamma \sum_i \log \mathcal{G}_{\mathbf{R}}(\mathcal{P}(\mathbf{R}_i)) + \beta\|\mathbf{R}\|_F^2, \tag{4}$$

where $\mathcal{G}_{\mathbf{R}}(\mathcal{P}(\mathbf{R}_i))$ is similar with $\mathcal{G}_{\mathbf{B}}(\mathcal{P}(\mathbf{B}_i))$. Note that we use two different GMMs for the background and rain layers, termed $\mathcal{G}_{\mathbf{B}}$ and $\mathcal{G}_{\mathbf{R}}$, respectively (we discuss further this GMM modeling in Sec. 3.3). Since the rain component tends to make up a small fraction of the observation, we impose $\|\mathbf{R}\|_F^2$ to penalize it, the importance of which is controlled by the parameter $\beta(=0.01)$.

Putting all terms together leads to the complete formulation of the energy function:

$$\begin{aligned} \min_{\mathbf{B},\mathbf{R}} \quad & \|\mathbf{O}-\mathbf{B}-\mathbf{R}\|_F^2 + \alpha\|\nabla\mathbf{B}\|_1 + \beta\|\mathbf{R}\|_F^2 - \\ & \gamma\sum_i \log\left(\mathcal{G}_{\mathbf{B}}(\mathcal{P}(\mathbf{B}_i)) + \log\mathcal{G}_{\mathbf{R}}(\mathcal{P}(\mathbf{R}_i))\right) \\ \text{s.t.} \quad & \forall i \;\; 0 \le \mathbf{B}_i, \mathbf{R}_i \le \mathbf{O}_i. \end{aligned} \tag{5}$$

The optimization approach to minimize this energy function is discussed in the next section.

3.2. Optimization

As noticed in Eq. (5), the cost function is non-convex due to the patch GMM priors. A commonly used scheme to solve this kind of problem is the half-quadratic splitting technique [8]. To cast our problem into the half-quadratic

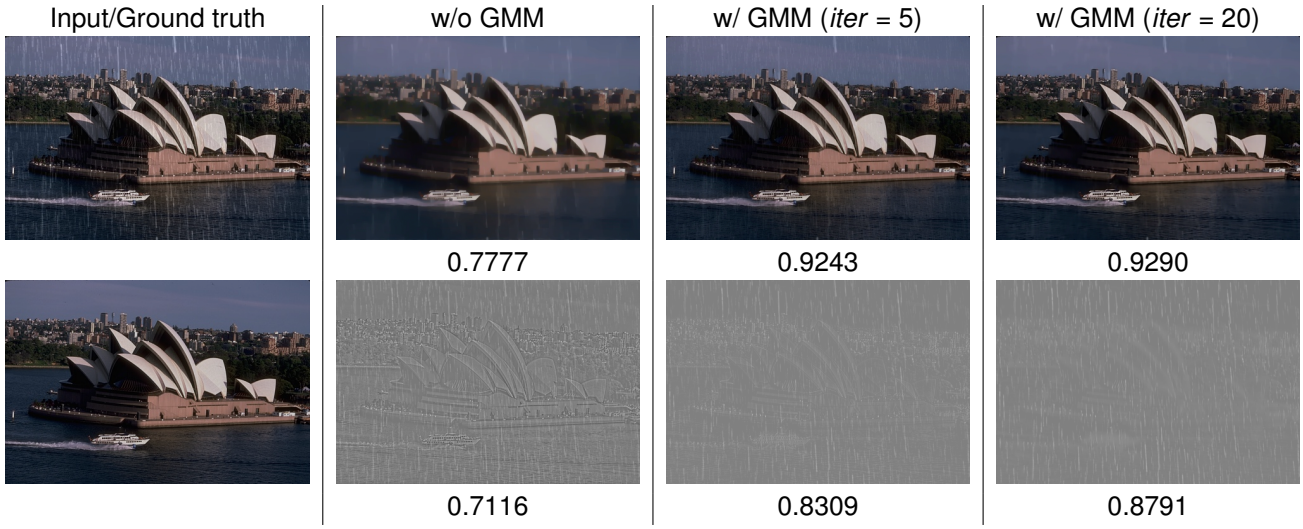

Figure 2: Illustration of the effect of the GMM. Our objective without the GMM components (second column) cannot distinguish the rain streaks and the image detains like the one with GMM (left two columns). The Structure Similarity Index (SSIM) for each result is shown below the image. Please refer to Sec. 4.1 for more analysis.

splitting framework, we need to make the objective function separable. Hence, auxiliary variables $\mathbf{g}_{\mathbf{B}_i}$, $\mathbf{g}_{\mathbf{R}_i}$ and $\mathbf{H}$ are introduced to replace $\mathcal{P}(\mathbf{B}_i)$, $\mathcal{P}(\mathbf{R}_i)$ and $\nabla\mathbf{B}$, respectively. By doing so, the optimization problem turns out to be in the following form:

$$\begin{aligned}\min\ & \|\mathbf{O}-\mathbf{B}-\mathbf{R}\|_F^2+\alpha\|\mathbf{H}\|_1+\beta\|\mathbf{R}\|_F^2- \\ & \gamma\sum_i\left(\log\mathcal{G}_{\mathbf{B}}(\mathbf{g}_{\mathbf{B}_i})+\log\mathcal{G}_{\mathbf{R}}(\mathbf{g}_{\mathbf{R}_i})\right)+\omega\|\nabla\mathbf{B}-\mathbf{H}\|_F^2 \\ & +\omega\sum_i\left(\|\mathcal{P}(\mathbf{B}_i)-\mathbf{g}_{\mathbf{B}_i}\|_2^2+\|\mathcal{P}(\mathbf{R}_i)-\mathbf{g}_{\mathbf{R}_i}\|_2^2\right) \\ & \text{s.t.}\quad \forall i\ \ 0\le\mathbf{B}_i,\mathbf{R}_i\le\mathbf{O}_i,\end{aligned} \tag{6}$$

where $\|\cdot\|_2$ represents the ℓ^2 norm. Notice that ω is a positive parameter that monotonically increases after each iteration. As ω grows, the solutions to Eq. (6) infinitely approach those to Eq. (5). The proposed algorithm iteratively updates the variables as described in the following.

Solving $\mathbf{H}$ Discarding the variables unrelated to $\mathbf{H}$ yields:

$$\mathbf{H}^{(t+1)}=\underset{\mathbf{H}}{\operatorname{argmin}}\,\alpha\|\mathbf{H}\|_1+\omega\|\nabla\mathbf{B}^{(t)}-\mathbf{H}\|_F^2. \tag{7}$$

This is a classic LASSO problem. Its closed-form solution can be efficiently obtained by the shrinkage operator, the definition of which on scalars is $\mathcal{S}_{\epsilon>0}[x]:=\operatorname{sgn}(x)\max(|x|-\epsilon,0)$. The extension of the shrinkage operator to vectors and matrices is simply applied element-wise. As a result, we have:

$$\mathbf{H}^{(t+1)}=\mathcal{S}_{\alpha/2\omega}[\nabla\mathbf{B}^{(t)}]. \tag{8}$$

Solving $\{\mathbf{B},\mathbf{R}\}$ By fixing $\mathbf{H}$, $\mathbf{g}_{\mathbf{B}_i}$ and $\mathbf{g}_{\mathbf{R}_i}$, the optimization problem corresponding to $\{\mathbf{B},\mathbf{R}\}$ becomes:

$$\begin{aligned}\{\mathbf{B}^{(t+1)},\mathbf{R}^{(t+1)}\}=\underset{\mathbf{B},\mathbf{R}}{\operatorname{argmin}}\ & \|\mathbf{O}-\mathbf{B}-\mathbf{R}\|_F^2+\beta\|\mathbf{R}\|_F^2+ \\ & \omega\sum_i\left(\|\mathcal{P}(\mathbf{B}_i)-\mathbf{g}_{\mathbf{B}_i}^{(t)}\|_2^2+\|\mathcal{P}(\mathbf{R}_i)-\mathbf{g}_{\mathbf{R}_i}^{(t)}\|_2^2\right) \\ & \text{s.t.}\quad \forall i\ \ 0\le\mathbf{B}_i,\mathbf{R}_i\le\mathbf{O}_i.\end{aligned} \tag{9}$$

Following [21], we use L-BFGS [28] to minimize this constraint L_2 problem.

Solving $\mathbf{g}_{\mathbf{B}_i}$ ($\mathbf{g}_{\mathbf{R}_i}$) Since the $\mathbf{g}_{\mathbf{B}_i}$ and $\mathbf{g}_{\mathbf{R}_i}$ sub-problems share the same formulation with the other variables given, we only detail the solver of $\mathbf{g}_{\mathbf{B}_i}$ here, while $\mathbf{g}_{\mathbf{R}_i}$ can be updated analogously. The optimization problem associated with each $\mathbf{g}_{\mathbf{B}_i^{(t+1)}}$ is expressed as:

$$\mathbf{g}_{\mathbf{B}_i}^{(t+1)}=\underset{\mathbf{g}_{\mathbf{B}_i}}{\operatorname{argmin}}\,\omega\|\mathcal{P}(\mathbf{B}_i^{(t+1)})-\mathbf{g}_{\mathbf{B}_i}\|_2^2-\gamma\log\mathcal{G}_{\mathbf{B}}(\mathbf{g}_{\mathbf{B}_i}). \tag{10}$$

It is worth mentioning that thanks to the independence of patches, the patches can be processed in parallel. The approximate optimization suggested by [29] is used. This approach applies Wiener filtering only using the component with the largest likelihood in the GMM. The whole process is summarized in Algorithm 1. Note that we first convert the RGB input image to the YUV space and remove rain streaks only on the luminance (Y) channel.

Algorithm 1 Rain Streak Removal Using Layer Priors

Input: input image $\mathbf{O}$; GMMs for two layers: $\mathcal{G}_\mathbf{B}$ and $\mathcal{G}_\mathbf{R}$;
Initialization: $\mathbf{B} \leftarrow \mathbf{O}$; $\mathbf{R} \leftarrow 0$; $\omega \leftarrow \omega_\circ$;
  repeat
    update $\mathbf{H}$ using Eq. 8;
    solve $\{\mathbf{B}, \mathbf{R}\}$ by Eq. 9;
    solve $\{\mathbf{g}_{\mathbf{B}_i}, \mathbf{g}_{\mathbf{R}_i}\}$ by Eq. 10;
    $\omega = 2 * \omega$;
  until convergence or maximum iteration number;
Output: The estimation of two layers $\mathbf{B}$ and $\mathbf{R}$;

3.3. Gaussian Mixture Model Learning

In this section, we describe how to obtain the two GMMs for the background and rain layers, namely $\mathcal{G}_\mathbf{B}$ and $\mathcal{G}_\mathbf{R}$. To obtain $\mathcal{G}_\mathbf{B}$, we utilize a pre-trained GMM model provided by [29] with 200 mixture components and patch size 8×8. This is learned from a set of 2×10^6 patches sampled from natural images, and thus can directly serve our purpose of modeling the background layer.

To obtain $\mathcal{G}_\mathbf{R}$, existing methods attempt to extract the internal properties of rain streaks within the input image itself, like [2, 11]. Similarly, we also learn the priors directly from the input image. Doing this guarantees the correctness of the rain streak appearance, which otherwise can be considerably different from one image to another image. Unlike [2, 11] that work on the entire image, we found that $\mathcal{G}_\mathbf{R}$ only requires small regions (*e.g.* a region with size 100×100 contains about $10K$ 8×8 patches inside), since rain streaks mostly form repetitive patterns. We observe that most natural images contain regions that are relatively flat, for instance, sky, building walls *etc*. The image patches within these regions can be approximated as pure rain streaks, and used to train $\mathcal{G}_\mathbf{R}$. This strategy of using local patches for global image restoration shares the similar spirit in [9].

To select such regions, we calculate the variance within each region in a sliding window fashion and pick the one with the least variance. Then, an EM algorithm is performed on the 8×8 patches sampled from the selected region to learn the parameters of $\mathcal{G}_\mathbf{R}$. We set the cluster number for $\mathcal{G}_\mathbf{R}$ to a small one, *i.e.* 20, compared with 200 for $\mathcal{G}_\mathbf{B}$, as the rain streak appearance of one single image has less variance than the background layer. Fig. 3 shows the eigenvectors of the three randomly selected mixture components from the learned $\mathcal{G}_\mathbf{R}$. Note that they have rain streak structures which contribute much to the expressive power of the model for the rain streak layer.

4. Experimental Results

We evaluate our method using both synthetic and real images, and compare our results with the state-of-the-art methods, including the sparse representation based dictionary learning method [11] (denoted as SR) and the low-rank appearance method [2] (denoted as LRA). For the experiments on synthetic data, the ground truth images are available, and we can evaluate and compare the results using the Structure Similarity Index (SSIM) [24] on the luminance channel, which indicates the closeness of the recovered background to the ground truth. Our Matlab implementation takes $93s$ (5 iterations) / $370s$ (20 iterations) to process one 480×640 color image.

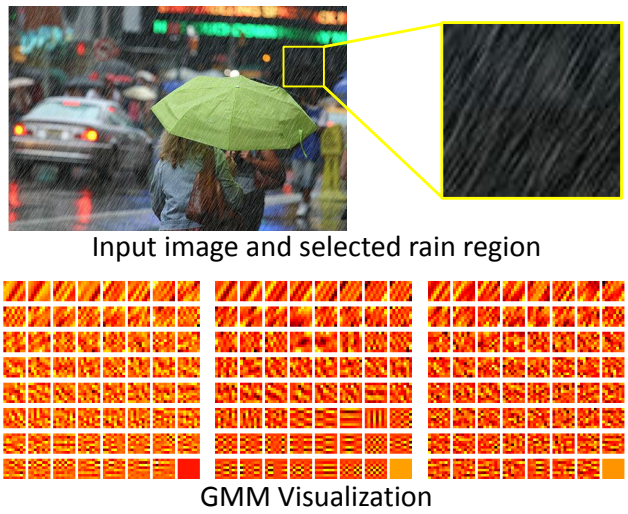

Figure 3: (up) The input image and the selected region used to learn the rain streak GMM. (bottom) Visualization of the eigenvectors of covariance of three randomly picked GMM components, sorted by their eigenvalues in descending order. This visualization helps to reveal that the GMM can capture the rain orientation and structure information.

4.1. Synthetic Data

Connection with the TV Model From the objective function in Eq. (5), one can notice that by simply disabling the terms related to $\mathbf{g}_{\mathbf{B}_i}$ and $\mathbf{g}_{\mathbf{R}_i}$, our model degenerates to the Total Variation (TV) model [20]. To reveal the benefit of the GMM priors, a comparison between our model (denoted as w/ GMM) and TV model (w/o GMM) is conducted. The effect is shown in Fig. 2, from which, we can see that TV is able to filter out the rain streaks but also falsely removes many details that belong to the background. Our method, on the other hand, is more effective at better recovering both the background and rain layers with only 5 iterations. In addition, the recovery quality can be continuously improved as the number of iterations increases. Quantitatively, our model achieves SSIM results (background: 0.9243, rain: 0.8309) with 5 iterations and reaches (0.9290, 0.8791) with 20 iterations, which are much better than those of TV (0.7777, 0.7116). In spite of the inferiority of TV model, its rough separation can be used as our starting point.

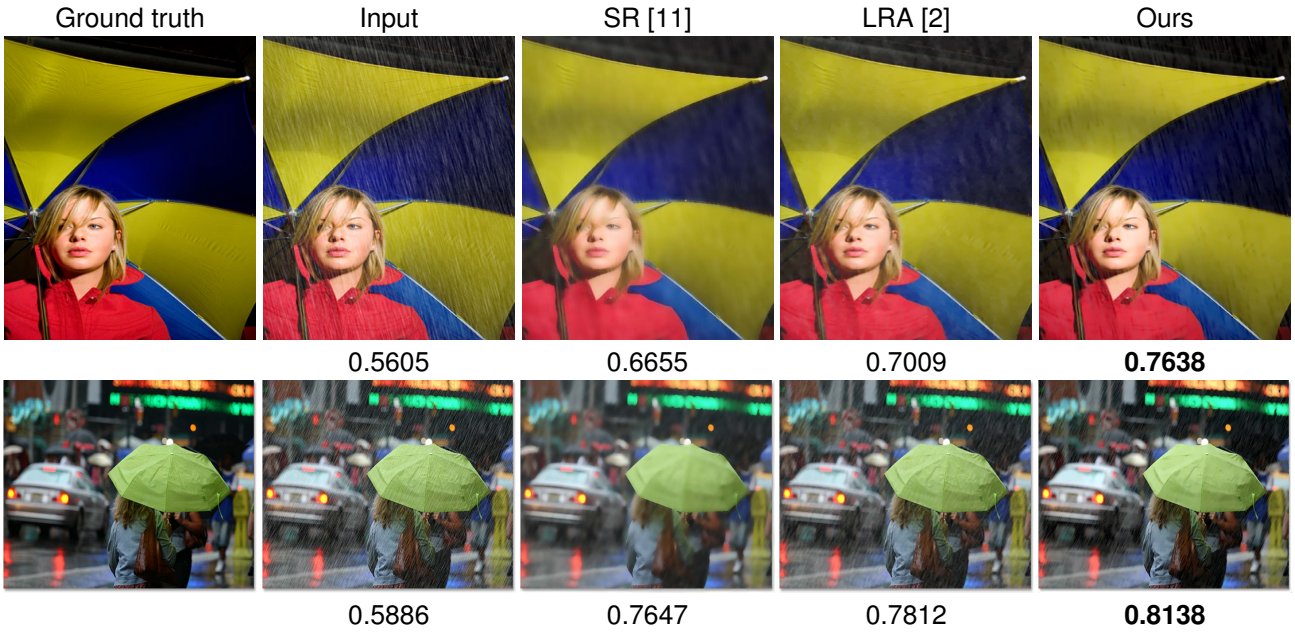

Figure 4: Rain streak removal results for the two datasets in [11]. Ours shows better background recovery both quantitatively and qualitatively.

Table 1: Quantitative comparison of rain streak removal results on our synthetic datasets (12 images) using SSIM.

SR[11]	0.73	0.79	0.83	0.76	0.62	0.73	0.82	0.77	0.73	0.74	0.63	0.76
LRA[2]	0.83	0.87	0.79	0.85	0.88	0.90	0.92	0.82	0.87	0.83	0.85	0.81
Ours	**0.88**	**0.93**	**0.93**	**0.93**	**0.90**	**0.95**	**0.96**	**0.90**	**0.91**	**0.90**	**0.86**	**0.92**

Comparisons Here we compare our method with LRA [2] and SR [11] methods. Figure 4 shows the results. As observed, our method considerably outperforms the other two methods in terms of both visual quality and SSIM. Compared with SR [11] that tends to over-smooth the image content and LRA [2] that sometimes fails to capture the rain streak layer, the proposed method removes the rain streaks while keeping more image details in the background layer.

In addition, we synthesize a new dataset with more images (12) using the photorealistic rendering techniques proposed by Grag and Shree [6]. Table 1 lists the performance of the three methods, the number in which confirm the noticeable advance of our methods. Figure 5 shows the visual results of two examples. Once again, the proposed method shows the best rain streak removal performance. Similar with the previous experiment, the defects of LRA and SR still exist. In other words, SR [11] again over-smooths the image content and cannot capture the rain streak in highly textured regions, while LRA [2] fails to remove the rain streaks in these two examples. The second example reveals another limitation using the low rank to model the rain streak layer likely because it may treat other repetitive patterns in the image as rain streaks *e.g.* the building windows.

4.2. Results on Real Images

Fig. 7 shows the results and comparisons using real images. Qualitatively, our method provides better results by retaining image details; while in these three cases, the other two methods tend to over-smooth the background.

When the rain is heavy, the density of rain streaks is so high that individual streaks cannot be observed anymore. For this case, we found that applying a dehazing method *e.g.* [17] as preprocessing is useful. Two examples are shown in Fig. 6, where combining the dehazing method and our rain streak removal can produce compelling results.

5. Conclusion

We have introduced a different approach to solve the decomposition problem of a background scene and rain streaks using a single image. Unlike existing methods, we impose constraints on both the background and rain layers. These constraints are simple patches learned using Gaussian mixture models (GMMs). Based on our experiments, we showed that these two constraints prove to be more effective than methods based on dictionary learning and low rank constraints.

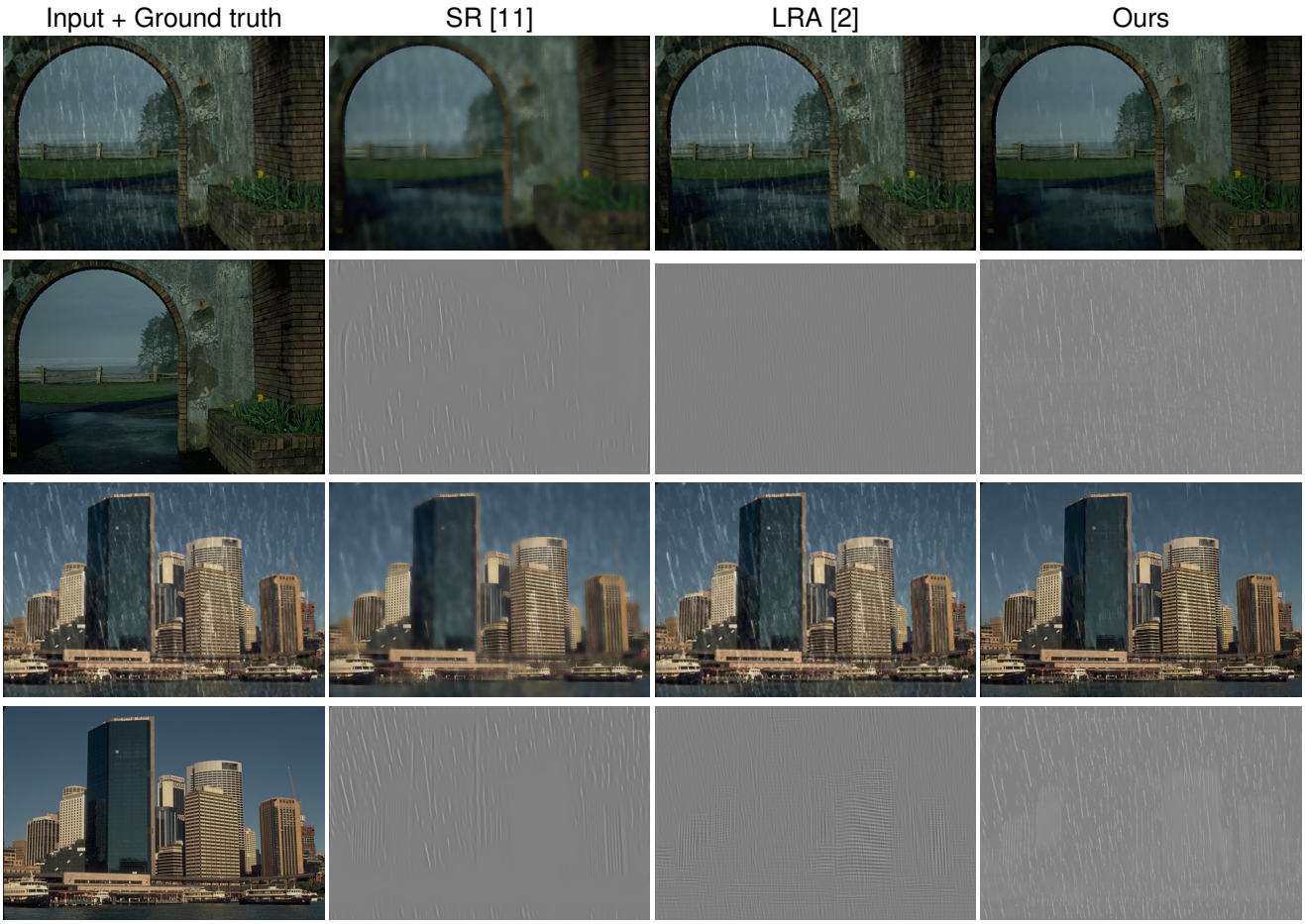

Figure 5: Visual comparison of different rain streak removal methods on a synthetic dataset.

Figure 6: Rain streak removal example for heavy rain. We found that first applying a dehazing method (3rd column), followed by the rain streak removal, helps improve results.

Figure 7: Visual comparison of different rain streak removal methods on real example images.

Our constraint on the rain layer is particularly interesting as rain streaks have special appearances and structures. GMM can effectively capture a considerably narrower distribution to describe rain streaks and distinguish them from the wider range of textures for the background layer. We have verified that this GMM prior for rain streaks is a critical part of the decomposition. Without the GMM priors, the estimated background is much more blurred, and the rain layer contains too much high-frequency texture from the background layer. In addition, the proposed method is not only simple and effective, but does not assume the rain streak orientations, sizes or scales. Thus, we consider demonstrating the usefulness of the GMM priors to the decomposition framework a step forward in addressing rain streak interference removal. Our method is not without limitations. Issues such as the optimal size and location of patches to estimate the GMMs, how to handle saturated rain pixels, and remaining rain artifacts in the output, are still open problems, which are targeted for future work.

Acknowledgement

This study is supported in part by the HCCS research grant at the ADSC from Singapore's Agency for Science, Technology and Research (A*STAR).

References

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

[2] Y.-L. Chen and C.-T. Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *IEEE Int'l Conf. Computer Vision*, 2013.

[3] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *IEEE Int'l Conf. Computer Vision*, 2013.

[4] K. Garg and S. K. Nayar. Detection and removal of rain from videos. In *IEEE Conf. Computer Vision and Pattern Recognition*, 2004.

[5] K. Garg and S. K. Nayar. When does a camera see rain? In *IEEE Int'l Conf. Computer Vision*, 2005.

[6] K. Garg and S. K. Nayar. Photorealistic rendering of rain streaks. *ACM Trans. Graphics*, 25(3):996–1002, 2006.

[7] K. Garg and S. K. Nayar. Vision and rain. *Int'l. J. Computer Vision*, 75(1):3–27, 2007.

[8] D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Trans. Image Processing*, 4(7):932–946, 1995.

[9] Z. Hu and M.-H. Yang. Good regions to deblur. In *European Conf. Computer Vision*. 2012.

[10] 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 Trans. Multimedia*, 16(1):83–93, 2014.

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

[12] J.-H. Kim, C. Lee, J.-Y. Sim, and C.-S. Kim. Single-image deraining using an adaptive nonlocal means filter. In *IEEE Int'l Conf. Image Processing*, 2013.

[13] H. Kurihata, T. Takahashi, I. Ide, Y. Mekada, H. Murase, Y. Tamatsu, and T. Miyahara. Rainy weather recognition from in-vehicle camera images for driver assistance. In *IEEE Intelligent Vehicles Symposium*, 2005.

[14] Y. Li and M. S. Brown. Exploiting reflection change for automatic reflection removal. In *IEEE Int'l Conf. Computer Vision*, 2013.

[15] Y. Li and M. S. Brown. Single image layer separation using relative smoothness. In *IEEE Conf. Computer Vision and Pattern Recognition*, 2014.

[16] Y. Luo, Y. Xu, and H. Ji. Removing rain from a single image via discriminative sparse coding. In *IEEE Int'l Conf. Computer Vision*, 2015.

[17] G. Meng, Y. Wang, J. Duan, S. Xiang, and C. Pan. Efficient image dehazing with boundary constraint and contextual regularization. In *IEEE Int'l Conf. Computer Vision*, 2013.

[18] M. Roser, J. Kurz, and A. Geiger. Realistic modeling of water droplets for monocular adherent raindrop recognition using bezier curves. In *Asian Conf. Computer Vision*, 2011.

[19] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *IEEE Conf. Computer Vision and Pattern Recognition*, 2005.

[20] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[21] Y. Shih, D. Krishnan, F. Durand, and W. T. Freeman. Reflection removal using ghosting cues. In *IEEE Conf. Computer Vision and Pattern Recognition*, 2015.

[22] S.-H. Sun, S.-P. Fan, and Y.-C. F. Wang. Exploiting image structural similarity for single image rain removal. In *IEEE Int'l Conf. Image Processing*, pages 4482–4486. IEEE, 2014.

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

[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 Processing*, 13(4):600–612, 2004.

[25] Y. Weiss and W. T. Freeman. What makes a good model of natural images? In *IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

[26] S. You, R. T. Tan, R. Kawakami, and K. Ikeuchi. Adherent raindrop modeling, detection and removal in video. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2015.

[27] 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 Int'l Conf. Multimedia and Expo*, 2006.

[28] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. on Mathematical Software*, 23(4):550–560, 1997.

[29] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *IEEE Int'l Conf. Computer Vision*, 2011.