Automatic Single-Image-Based Rain Streaks Removal via Image Decomposition

Li-Wei Kang, *Member, IEEE*, Chia-Wen Lin, *Senior Member, IEEE*, and Yu-Hsiang Fu

***Abstract*—Rain removal from a video is a challenging problem and has been recently investigated extensively. Nevertheless, the problem of rain removal from a single image was rarely studied in the literature, where no temporal information among successive images can be exploited, making the problem very challenging. In this paper, we propose a single-image-based rain removal framework via properly formulating rain removal as an image decomposition problem based on morphological component analysis. Instead of directly applying a conventional image decomposition technique, the proposed method first decomposes an image into the low- and high-frequency (HF) parts using a bilateral filter. The HF part is then decomposed into a "rain component" and a "nonrain component" by performing dictionary learning and sparse coding. As a result, the rain component can be successfully removed from the image while preserving most original image details. Experimental results demonstrate the efficacy of the proposed algorithm.**

***Index Terms*—Dictionary learning, image decomposition, morphological component analysis (MCA), rain removal, sparse representation.**

I. Introduction

DIFFERENT weather conditions such as rain, snow, haze, or fog will cause complex visual effects of spatial or temporal domains in images or videos [1]–[11]. Such effects may significantly degrade the performances of outdoor vision systems relying on image/video feature extraction [12]–[17] or visual attention modeling [18], such as image registration [10], event detection [9], object detection [15]–[17], tracking, and recognition, scene analysis [18] and classification, image indexing and retrieval [12], and image copy/near-duplicate detection. A comprehensive survey of detection approaches for outdoor environmental factors such as rain and snow to enhance the accuracy of video-based automatic incident detection systems can be found in [9].

Removal of rain streaks has recently received much attention [2]–[4], [6]. To the best of our knowledge, current approaches are all based on detecting and removing rain streaks in a video. This paper is among the first to specifically address the problem of removing rain streaks in a single image. Note that rain removal in an image may also fall into the category of the problem about image noise removal or image restoration. Hence, in the following subsections, we first briefly review current vision-based (video-based) rain removal approaches and image noise removal, followed by presenting our motivations of single-image-based rain streak removal and contribution of the proposed method.

A. Vision-Based Rain Detection and Removal

A pioneering work on detecting and removing rain streaks in a video was proposed in [2], where the authors developed a correlation model capturing the dynamics of rain and a physics-based motion blur model characterizing the photometry of rain. It was subsequently shown in [3] that some camera parameters such as exposure time and depth of field can be selected to mitigate the effects of rain without altering the appearance of the scene. Moreover, an improved video rain streak removal algorithm incorporating both temporal and chromatic properties was proposed in [6]. The method proposed in [7] further utilizes the shape characteristics of rain streak for identifying and removing rain streaks from videos. Furthermore, a model of the shape and appearance of a single rain or snow streak in the image space was developed in [1] to detect rain or snow streaks. Then, the amount of rain or snow in the video can be reduced or increased. In [8], selection rules based on photometry and size are proposed to select the potential rain streaks in a video, where a histogram of orientations of rain streaks, estimated with geometric moments, is computed.

Moreover, some research works [10], [11] focus on raindrop detection in images or videos (usually on car windshields) that is different from the detection of rain streaks. A video-based raindrop detection method for improving the accuracy of image registration was proposed in [10], where a photometric raindrop model was utilized to perform monocular raindrop detection in video frames. In addition, a detection method for detecting raindrops on car windshields using geometric–photometric environment construction and intensity-based correlation was proposed in [11], which can be applied to vision-based driver assistance systems.

B. Image Noise Removal

Image noise removal or denoising problem is important and challenging [19]. The major goal of image noise removal is to design an algorithm that can remove unstructured or structured noise from an image, which is acquired in the presence of an additive noise. Numerous contributions for image denoising in

Manuscript received March 19, 2011; revised August 26, 2011 and November 16, 2011; accepted November 17, 2011. Date of publication December 09, 2011; date of current version March 21, 2012. This work was supported in part by the National Science Council, Taiwan, under Grants NSC98-2221-E-007-080-MY3, NSC100-2218-E-001-007-MY3, and NSC100-2811-E-001-005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Brendt Wohlberg.

L.-W. Kang is with Institute of Information Science, Academia Sinica, Taipei 11529, Taiwan.

C.-W. Lin is with the Department of Electrical Engineering and the Institute of Communications Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan (e-mail: cwlin@ee.nthu.edu.tw).

Y.-H. Fu is with MStar Semiconductor Inc., Hsinchu 302, Taiwan.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2011.2179057

the past 50 years addressed this problem from many and diverse points of view. For example, spatial adaptive filters, stochastic analysis, partial differential equations, transform-domain methods, splines, approximation theory methods, and order statistics are some of the directions explored to address this problem [20]. Recently, the use of sparse and redundant representations over learned dictionaries has become one specific approach toward image denoising, which has been proven to be effective and promising [20]. Based on the assumption that image signals admit a sparse decomposition over a redundant dictionary, by using the K-SVD dictionary training algorithm [21], Elad and Aharon [20] obtained a dictionary describing the image content effectively. They proposed two training options, where one is using the corrupted image itself and the other one is training on a set of high-quality images. They have shown how such Bayesian treatment leads to a simple and effective denoising algorithm, which achieves state-of-the-art image denoising performance. Moreover, a similar idea has been successfully extended to solve more general image restoration problems such as removing nonhomogeneous noise or recovering missing information (e.g., text removal and inpainting [22], [23] and binary artifacts removal from video game images [24]). Although these dictionary-based image denoising methods can be also used for removing rain streaks, they usually cannot do a good job in rain removal, as will be shown in Section V.

C. Motivations of Single-Image-Based Rain Streak Removal

So far, the research works on rain streak removal found in the literature have been mainly focused on video-based approaches that exploit temporal correlation in multiple successive frames. Nevertheless, when only a single image is available, such as an image captured from a digital camera/camera phone or downloaded from the Internet, a single-image-based rain streak removal approach is required, which was rarely investigated before. In addition, some video rain removal approaches [3] based on adjusting camera parameters may not be suitable to consumer camcorders [6] and cannot be applied to existing acquired image/video data. Furthermore, for removing rain streaks from videos acquired from a moving camera, the performances of existing video-based approaches may be significantly degraded. The reason is that, since these video-based approaches usually perform rain streak detection, followed by interpolating the detected pixels affected by rain streaks in each frame, the nonstationary background due to camera motions and inaccurate motion estimation caused by the interference of rain streaks would degrade the accuracy of video-based rain streak detection and pixel interpolation. Although some camera motion estimation techniques can be applied first to compensate for the camera motions [6], its performance may be also degraded by rain streaks or large moving activity. Moreover, for the case of steady effects of rain, i.e., pixels may be affected by rain across multiple consecutive frames, it is hard to detect these pixels or find reliable information from neighboring frames to recover them [2].

Moreover, many image-based applications such as mobile visual search [12], object detection/recognition, image registration, image stitching, and salient region detection heavily rely on extraction of gradient-based features that are rotation and scale invariant. Some widely used features (descriptors) such as scale-invariant feature transform (SIFT) [13], speeded up robust features (SURFs) [14], and histogram of oriented gradients (HOGs) [15]–[17] are mainly based on computation of image gradients. The performances of these gradient-based feature extraction schemes, however, can be significantly degraded by rain streaks appearing in an image since the rain streaks introduce additional time-varying gradients in similar directions. For example, as illustrated in Fig. 1, the additional unreliable interesting points caused by rain streaks degrade the invariant properties of SIFT/SURF and lead to potentially erroneous image matching in related applications. As an example shown in Fig. 2, we applied the HOG-based pedestrian detector released from [17] to the rain image shown in Fig. 2(a) and its rain-removed version (obtained by the proposed method presented in Section III) shown in Fig. 2(b). It can be found that the detection accuracy for the rain-removed version is better. In

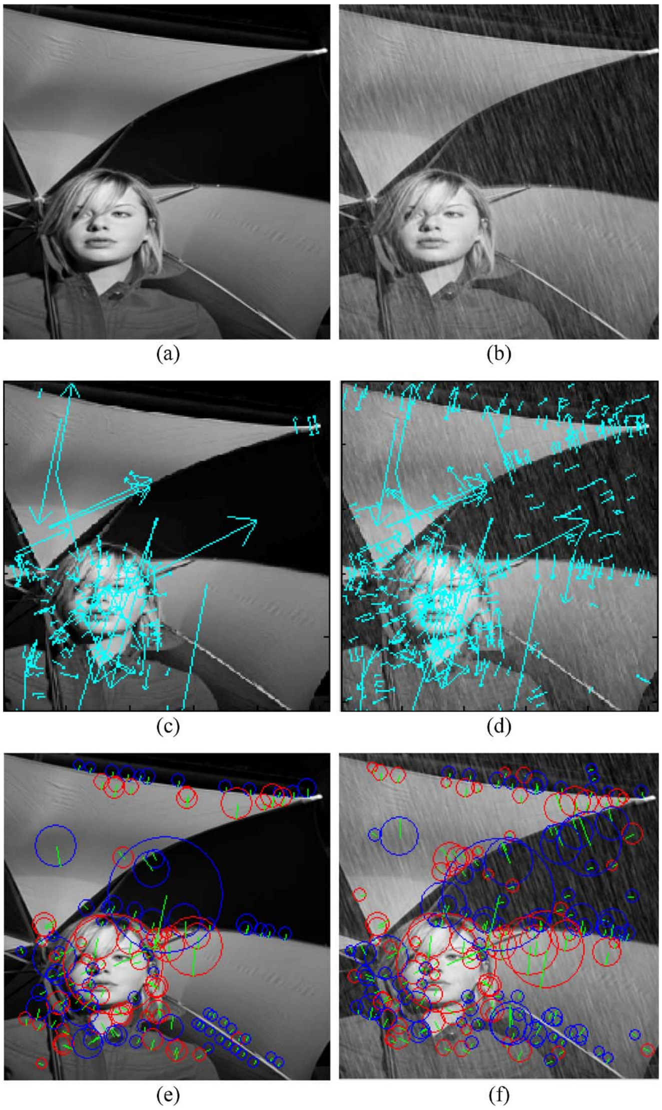
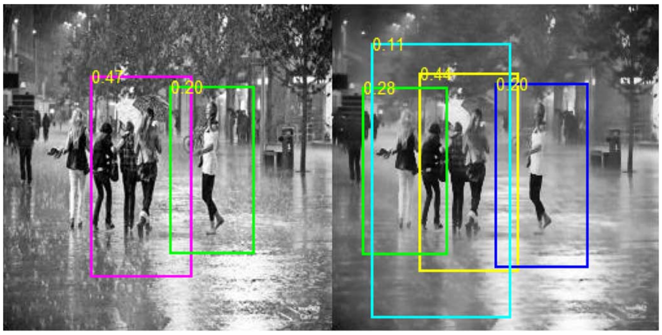


Fig. 1. Examples of interesting point detection: (a) original nonrain image; (b) rain image of (a); (c) SIFT interesting point detection for (a) (169 points); (d) SIFT interesting point detection for (b) (421 points); (e) SURF interesting point detection for (a) (131 points); and (f) SURF interesting point detection for (b) (173 points).



(a) (b)

Fig. 2. Applying the HOG-based pedestrian detector released from [17] to: (a) original rain image (four pedestrians detected) and (b) rain-removed version (obtained by the proposed method) of (a) (five pedestrians detected).

addition, visual attention models [18] compute a saliency map topographically encoding for saliency at each location in the visual input that simulates which elements of a visual scene are likely to attract the attention of human observers. Nevertheless, the performances of the model for related applications may be also degraded if rain streaks directly interact with the interested target in an image. Therefore, single-frame-based rain streak removal is desirable.

D. Contribution of the Proposed Method

It should be noted that separating and removing rain streaks from the nonrain part in a single frame is not a trivial work as rain streaks are usually highly mixed with the nonrain part, making the decomposition of the nonrain part very challenging. In this paper, we propose a single-image-based rain streak removal framework by formulating rain streak removal as an image decomposition problem based on morphological component analysis (MCA) [26]–[30]. In our method, an image is first decomposed into the low-frequency (LF) and high-frequency (HF) parts using a bilateral filter. The HF part is then decomposed into “rain component” and “nonrain component” by performing dictionary learning and sparse coding based on MCA. The major contribution of this paper is threefold: 1) to the best of our knowledge, our method is among the first to achieve rain streak removal while preserving geometrical details in a single frame, where no temporal or motion information among successive images is required; 2) we propose the first automatic MCA-based image decomposition framework for rain steak removal; and 3) the learning of the dictionary for decomposing rain steaks from an image is fully automatic and self-contained, where no extra training samples are required in the dictionary learning stage. In addition, the proposed method also offers another option in dictionary learning by collecting exemplar patches from a set of nonrain training images to learn an extended dictionary to enrich the dictionary, as detailed in Section III-C.

The rest of this paper is organized as follows. In Section II, we briefly review the concepts of MCA-based image decomposition, sparse coding, and dictionary learning techniques. Section III presents the proposed single-image-based rain streak removal framework. In Section IV, experimental results are demonstrated. Finally, Section V concludes this paper.

TABLE I
NOTATION

Symbols	Meanings
I	Input rain image
I^{Non_Rain}	Rain-removed version of I
N	Number of patches in I
S	Number of layers (or morphological components) in I
I_s	The s-th component decomposed from I
b_s^k	The k-th image patch extracted from I_s
θ_s, θ_s^k	Sparse coefficients of I_s and b_s^k
D_s	Dictionary for sparsely representing I_s
D_{gs}, D_{ls}	Global and local dictionaries for sparsely representing I_s
λ, τ	Regularization parameters
M_s, m_s	Numbers of elements of θ_s and θ_s^k
y^k, θ^k	The k-th patch training exemplar and its sparse coefficients
I_{LF}, I_{HF}	Low-frequency and high-frequency parts of I
I_{HF}^G, I_{HF}^R	Geometric component and rain component of I_{HF}
D_{HF}	Dictionary learned from the training exemplars extracted from I_{HF}
D_{HF_G}, D_{HF_R}	Sub-dictionaries divided from for sparsely representing I_{HF}^G and I_{HF}^R
D_{EG}	Extended global dictionary
b_{HF}^k	The k-th patch extracted from I_{HF}
θ_{HF}^k	Sparse coefficients of b_{HF}^k
$\tilde{\theta}_{HF}^k$	Reconstructed sparse coefficients of b_{HF}^k
$\tilde{\theta}_{HF\_G}^k, \tilde{\theta}_{HF\_R}^k$	Sparse coefficients in $\tilde{\theta}_{HF}^k$ corresponding to D_{HF_G} and D_{HF_R}
$\tilde{b}_{HF\_G}^k, \tilde{b}_{HF\_R}^k$	The k-th reconstructed geometric patch and rain patch
$\mu(D_1, D_2)$	Mutual coherence between dictionaries D_1 and D_2

II. MCA-Based Image Decomposition, Sparse Coding, and Dictionary Learning

The key idea of MCA is to utilize the morphological diversity of different features contained in the data to be decomposed and to associate each morphological component to a dictionary of atoms. Here, the conventional MCA-based image decomposition approaches [26]–[30], sparse coding [31], [32], and dictionary learning [21], [33] techniques are briefly introduced. The symbols used in this paper are listed in Table I.

A. MCA-Based Image Decomposition

Suppose that an image I of N pixels is a superposition of S layers (called morphological components), denoted by $I = \sum_{s=1}^{S} I_s$, where I_s denotes the sth component, such as the geometric or textural component of I. To decompose the image I into $\{I_s\}_{s=1}^{S}$, the MCA algorithms [26]–[30] iteratively minimize the following energy function:

$$E\left(\{I_s\}_{s=1}^{S}, \{\theta_s\}_{s=1}^{S}\right) = \frac{1}{2}\left\| I - \sum_{s=1}^{S} I_s \right\|_2^2 + \tau \sum_{s=1}^{S} E_s(I_s, \theta_s) \tag{1}$$

where $\theta \in R^{M_s}$ denotes the sparse coefficients corresponding to I_s with respect to dictionary D_s, τ is a regularization parameter, and E_s is the energy defined according to the type of D_s (denoted by D_{gs} for a global dictionary or by D_{ls} for a local dictionary). For a global dictionary $D_{gs} \in R^{N \times M_s}$, $N \leq M_s$, energy function E_s is defined as

$$E_s(I_s, \theta_s) = \frac{1}{2}\|I_s - D_{gs}\theta_s\|_2^2 + \lambda\|\theta_s\|_1 \tag{2}$$

where τ is a regularization parameter. Usually, to decompose an image into its geometric and textural components, traditional basis functions such as wavelets or curvelets are used as the dictionary for representing the geometric component, whereas global discrete cosine transform (DCT) basis functions are used as the dictionary for representing the textural component of the image [26]–[30].

With respect to a local dictionary $D_{sl} \in R^{n \times m_s}$, $n \leq m_s$, $\theta_s^k \in R^{m_s}$ represents the sparse coefficients of patch $b_s^k \in R^n$, $k = 1, 2, \ldots, N$, extracted from I_s. Each patch b_s^k can be extracted centralized with a pixel of I_s and overlapped with adjacent patches. The energy function E_s for the local dictionary can be defined as

$$E_s(I_s, \theta_s) = \sum_{k=1}^{N} \left(\frac{1}{2} \left\| b_s^k - D_{ls}\theta_s^k \right\|_2^2 + \lambda \left\| \theta_s^k \right\|_1 \right). \qquad (3)$$

Usually, a local dictionary for representing the textural component of an image is either composed of traditional basis functions, such as local DCT [26]–[28], [30], or constructed from the dictionary learning procedure [29] described in Section II-B.

The MCA algorithms solve (1) by iteratively performing for each component I_s the following two steps: 1) update of the sparse coefficients, i.e., this step performs sparse coding to solve θ_s or $\{\theta_s^k\}_{k=1}^N$ to minimize $E_s(I_s, \theta_s)$ while fixing I_s; and 2) update of the components, i.e., this step updates I_s or $\{b_s^k\}_{k=1}^N$ while fixing θ_s or $\{\theta_s^k\}_{k=1}^N$.

More specifically, in the case of decomposing I into two components I_s, $s = 1, 2$, a key step of MCA is to properly select a dictionary built by combining two subdictionaries D_s, $s = 1, 2$, D_1 and D_2 that can be either global or local dictionaries and should be mutually incoherent, i.e., D_1 can provide sparse representation for I_1, but not for I_2, and vice versa. To decompose I into geometric I_1 and textural I_2 components, global wavelet or global curvelet is used as D_1, whereas global DCT or local DCT is used as D_2 in [26]–[28] and [30]. A comprehensive description of dictionary selections and related parameter settings for different kinds of image decomposition can be found in Table 2 in [27]. On the other hand, in [29], a global wavelet/curvelet basis is also used as D_1, whereas D_2 is constructed through a local dictionary learning process described below. Finally, to decompose an image into two components, both D_1 and D_2 are required to sparsely represent each component individually, as illustrated in Fig. 3, for the proposed single-image-based rain streak removal. It should be noted that we do not directly apply (1), (2), and (3) to solve the rain streak removal problem. The major differences between the proposed framework and traditional MCA-based approaches are described in Section III-A. More details about traditional MCA methods, such as parameter settings, can be found in [26]–[30].

B. *Sparse Coding and Dictionary Learning*

Sparse coding [31], [32] is a technique of finding a sparse representation for a signal with a small number of nonzero or significant coefficients corresponding to the atoms in a dictionary [21], [33]. The pioneering work in sparse coding proposed by Olshausen and Field [31] states that the receptive fields of simple cells in mammalian primary visual cortex can be characterized as being spatially localized, oriented, and bandpass. It was shown in [31] that a coding strategy that maximizes sparsity is sufficient to account for these three properties and that a learning algorithm attempting to find sparse linear codes for natural scenes will develop a complete family of localized, oriented, and bandpass receptive fields.

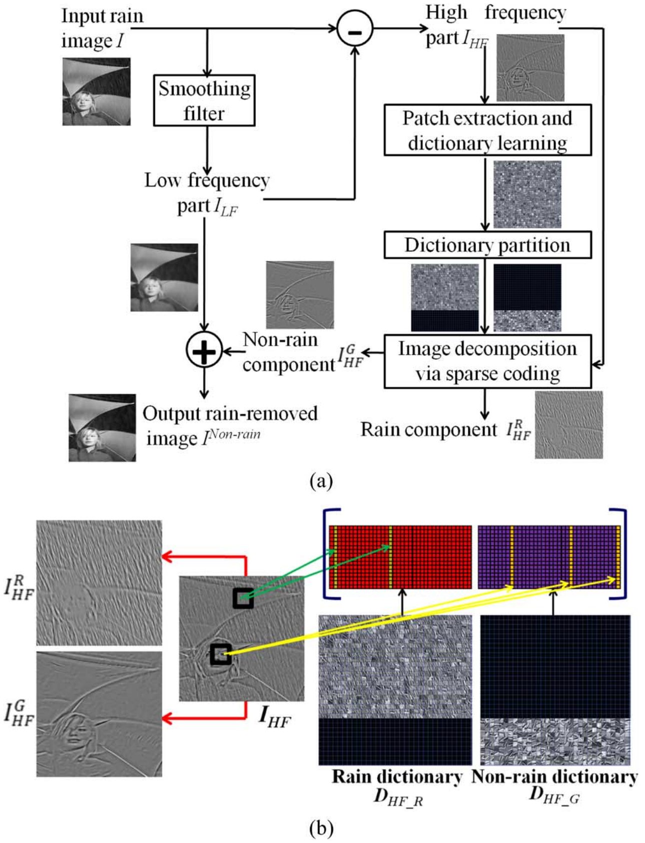

Input rain image I
Smoothing filter
Low frequency part I_{LF}
High frequency part I_{HF}
Patch extraction and dictionary learning
Dictionary partition
Image decomposition via sparse coding
Non-rain component I_{HF}^G
Output rain-removed image $I^{Non\text{-}rain}$
Rain component I_{HF}^R
(a)
I_{HF}^R
I_{HF}^G
I_{HF}
Rain dictionary D_{HF_R}
Non-rain dictionary D_{HF_G}
(b)

Fig. 3. (a) Block diagram of the proposed rain streak removal method. (b) Illustration of the proposed method based on two learned local dictionaries.

As aforementioned, it is required to construct a dictionary D_{ls} containing the local structures of textures for sparsely representing each patch b_s^k extracted from the textural component I_s of image I. In some applications, we may use a set of available training exemplars (similar to the patches extracted from the component we want to decompose) $y^k \in R^n$, $k = 1, 2, \ldots, P$, to learn a dictionary D_{ls} sparsifying y^k by solving the following optimization problem:

$$\min_{D_{ls} \in R^{n \times m_s}, \theta^k \in R^{m_s}} \sum_{k=1}^{P} \left(\frac{1}{2} \left\| y^k - D_{ls}\theta^k \right\|_2^2 + \lambda \|\theta^k\|_1 \right) \qquad (4)$$

where θ^k denotes the sparse coefficients of y^k with respect to D_{ls}, and λ is a regularization parameter. Equation (4) can be efficiently solved by performing a dictionary learning algorithm, such as K-SVD [21] or online dictionary learning [33] algorithms, where the sparse coding step is usually achieved via orthogonal matching pursuit (OMP) [32]. Finally, image decomposition is achieved by iteratively performing the MCA algorithm to solve I_s (while fixing D_{ls}) described in Section II-A and the dictionary learning algorithm to learn D_{ls} (while fixing I_s) until convergence. The convergence of the MCA image decomposition algorithms has been proven in [29].

The proposed rain removal framework described in Section III uses two local dictionaries learned from the training patches extracted from the rain image itself to respectively decompose a rain image into its rain component and geometric (nonrain) component without using any global dictionary. The main reasons include: 1) we do not assume or empirically decide any type of global dictionary for representing either of the rain and geometrical components in the rain image; 2) because the geometric component is usually highly mixed with rain streaks in some regions of the rain image, segmenting the image into local patches would be easier to extract rain patches that mainly contain rain streaks to facilitate self-learning of rain atoms; and 3) since rain streaks in different local regions of an image often exhibit different characteristics, local-patch-based dictionary learning would usually learn rain atoms that better represent rain streaks than a global dictionary does.

III. Proposed Rain Streak Removal Framework

Fig. 3 shows the proposed single-image-based rain streak removal framework, in which rain streak removal is formulated as an image decomposition problem. In our method, the input rain image is first roughly decomposed into the LF and HF parts using the bilateral filter [34], [35], where the most basic information will be retained in the LF part whereas the rain streaks and the other edge/texture information will be included in the HF part of the image, as illustrated in Fig. 4(a) and (b). Then, we perform the proposed MCA-based image decomposition to the HF part that can be further decomposed into the rain component [see Fig. 4(c)] and the geometric (nonrain) component [see Fig. 4(d)]. In the image decomposition step, a dictionary learned from the training exemplars extracted from the HF part of the image itself can be divided into two subdictionaries by performing HOG [15] feature-based dictionary atom clustering. Then, we perform sparse coding [32] based on the two subdictionaries to achieve MCA-based image decomposition, where the geometric component in the HF part can be obtained, followed by integrating with the LF part of the image to obtain the rain-removed version of this image, as illustrated in Fig. 4(e) and (f). The detailed method shall be elaborated below.

A. Major Differences Between the Proposed Method and Traditional MCA-Based Approaches

As mentioned in Section II, traditional MCA algorithms usually use a fixed global dictionary based on wavelets/curvelets to represent the geometric component of an image. To represent the textural component of an image, either a fixed global (global DCT) or a local (local DCT) dictionary is used. In addition, a learned dictionary may be also used to represent the textural component. Nevertheless, to decompose an image into the geometric and textural components, the selection of dictionaries and related parameter tuning seems to be heavily empirical, as the examples shown in Table 2 in [27]. Based on our experience, it is not easy to select a proper fixed dictionary to represent rain streaks due to its variety.

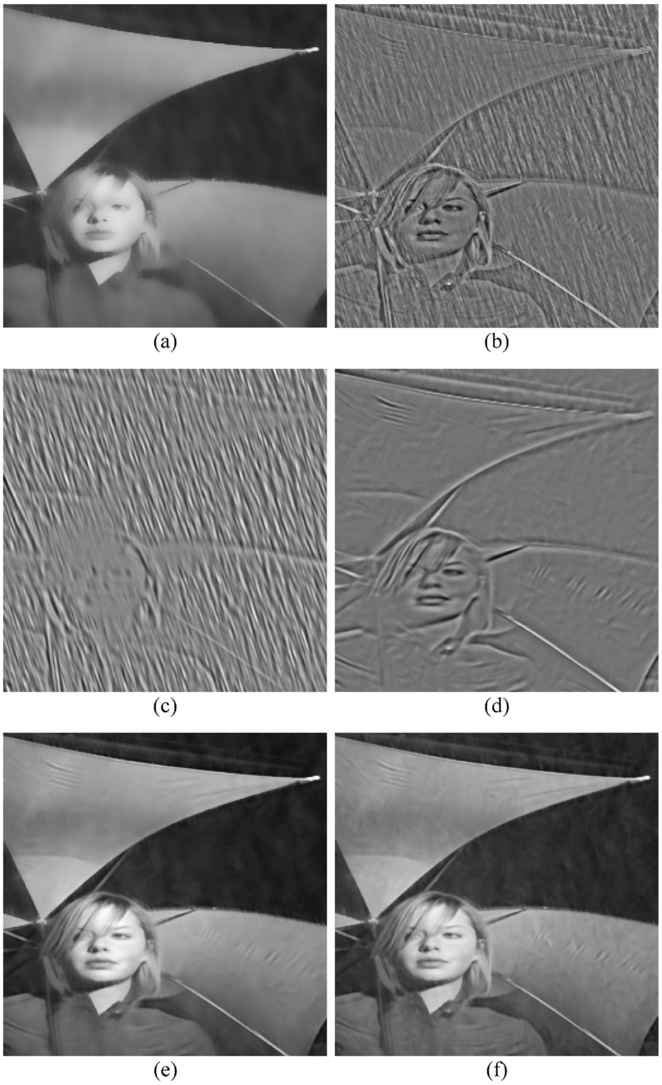


Fig. 4. Step-by-step results of the proposed rain streak removal process: (a) the LF part of the rain image in Fig. 1(b) decomposed using the bilateral filter (VIF $= 0.33$); (b) HF part; (c) rain component; and (d) geometric component. Combining (d) and the LF part shown in (a) to obtain: (e) the rain-removed version for the rain image shown in Fig. 1(b) (VIF $= 0.50,\ \mu = 0.6970$); (f) the rain-removed version for the rain image shown in Fig. 1(b) with D_E (VIF $= 0.52$).

Moreover, learning a dictionary for representing textural component usually assumes that a set of exemplar patches for the texture to be represented can be either known in advance or extracted from an image to be decomposed itself. Nevertheless, in practice, it is usually not easy to select correct rain patches in a single rain image automatically. It is also not easy to directly extract pure rain patches for dictionary learning from a rain image because rain streaks usually cover most regions in a rain image. That is, the geometric and rain components are usually largely mixed. Moreover, although a traditional fixed global dictionary based on wavelets/curvelets can well sparsely represent the geometric component of an image, using a learned dictionary based on the exemplar patches extracted from the component itself would be much better [38].

Therefore, rather than using a fixed dictionary, assuming prior training exemplar patches available, or resorting to tuning parameters for the used dictionary, our method extracts a set of selected patches from the HF part of a rain image itself to learn a dictionary. Then, based on the features extracted from individual atoms, we classify the atoms constituting the dictionary into two clusters to form two subdictionaries for representing the geometric and rain components of the image. The dictionary learning process in the proposed method is elaborated in Section III-C.

Traditional MCA algorithms are all directly performed on an image in the pixel domain. However, it is typically not easy to directly decompose an image into its geometric and rain components in the pixel domain because the geometric and rain components are usually largely mixed in a rain image. This makes the dictionary learning process difficult to clearly identify the "geometric (nonrain) atoms" and "rain atoms" from the pixel-domain training patches with mixed components. This may lead to removing too many image contents that belong to the geometric component but are erroneously classified to the rain component.

Therefore, we propose to first roughly decompose a rain image into the LF and HF parts. Obviously, the most basic information of the image is retained in the LF part, whereas the rain component and the other edge/texture information are mainly included in the HF part. The decomposition problem can be therefore converted to decomposing the HF part into the rain and other textural components. Such decomposition aids in the dictionary learning process as it is easier to classify in the HF part "rain atoms" and "nonrain atoms" into two clusters based on some specific characteristics of rain streaks.

Furthermore, traditional MCA-based image decomposition approaches are all achieved by iteratively performing the MCA algorithm and the dictionary learning algorithm until convergence. In contrast, the proposed method is noniterative except for that the utilized dictionary learning, clustering, and sparse coding tools are essentially iterative, as will be explained below.

B. Preprocessing and Problem Formulation

For an input rain image I, in the preprocessing step, we apply a bilateral filter [34] to roughly decompose I into the LF part I_{LF}) and the HF part I_{HF}, i.e., $I = I_{\mathrm{LF}} + I_{\mathrm{HF}}$. The bilateral filter can smooth an image while preserving edges by means of a nonlinear combination of nearby image values. In this step, we adjust the strength of smoothness of the bilateral filter to remove all of the rain streaks from I, as an illustrative example shown in Fig. 4(a) and (b). Then, our method learns a dictionary D_{HF} based on the training exemplar patches extracted from I_{HF} to further decompose I_{HF}, where D_{HF} can be further divided into two subdictionaries, i.e., $D_{\mathrm{HF\_}G}$ and $D_{\mathrm{HF\_}R}(D_{\mathrm{HF}} = [D_{\mathrm{HF\_}G}|D_{\mathrm{HF\_}R}])$, for representing the geometric and rain components of I_{HF}, respectively. As a result, we formulate the problem of rain streak removal for image I as a sparse coding-based image decomposition problem as follows:

$$\min_{\theta_{\mathrm{HF}}^k \in R^m} \left\| b_{\mathrm{HF}}^k - D_{\mathrm{HF}} \theta_{\mathrm{HF}}^k \right\|_2^2 \text{ s.t. } \left\| \theta_{\mathrm{HF}}^k \right\|_0 \leq L \tag{5}$$

where $b_{\mathrm{HF}}^k \in R^n$ represents the kth patch extracted from I_{HF}, $k = 1, 2, \ldots, P$. $\theta_{\mathrm{HF}}^k \in R^m$ are the sparse coefficients of b_{HF}^k with respect to $D_{\mathrm{HF}} \in R^{n\times m}$, $n \leq m$, and L denotes the sparsity or maximum number of nonzero coefficients of θ_{HF}^k. Since l_0 minimization is hard to optimize, one usually solves the l_1-minimization problem, which, in most cases, gives comparable results [36], [37]. Therefore, solving the l_0-minimization problem in (5) can be cast to solve the following l_1-minimization problem:

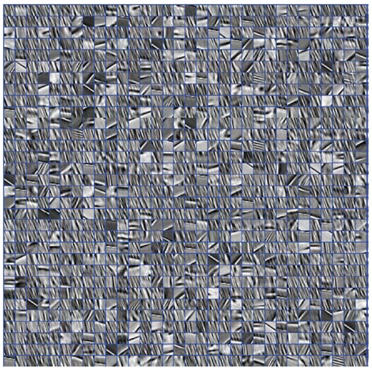


Fig. 5. Dictionary learned from the patches extracted from the HF part shown in Fig. 4(b) via the online dictionary learning for sparse coding algorithm [33], where each atom is of size 16 × 16.

$$\left(\theta_{\mathrm{HF}}^k\right)^* = \arg \min_{\theta_{\mathrm{HF}}^k \in R^m} \left(\frac{1}{2} \left\| b_{\mathrm{HF}}^k - D_{\mathrm{HF}} \theta_{\mathrm{HF}}^k \right\|_2^2 + \lambda \left\| \theta_{\mathrm{HF}}^k \right\|_1 \right) \tag{6}$$

where $(\theta_{\mathrm{HF}}^k)^*$ denotes the solution minimizing (6), and λ is a regularization parameter. To solve (5), we apply the efficient OMP implementation provided in [33]. Each patch b_{HF}^k can be reconstructed and used to recover either the geometric or rain component of I_{HF} depending on the corresponding nonzero coefficients in θ_{HF}^k, i.e., the used atoms from $D_{\mathrm{HF\_}G}$ or $D_{\mathrm{HF\_}G}$.

C. Dictionary Learning and Partition

Dictionary Learning: In this step, we extract from I_{HF} a set of overlapping patches as the training exemplars y^k for learning dictionary D_{HF}. We formulate the dictionary learning problem as [21], [33]

$$\min_{D_{\mathrm{HF}} \in R^{n\times m}, \theta^k \in R^m} \frac{1}{P} \sum_{k=1}^{P} \left(\frac{1}{2} \| y^k - D_{\mathrm{HF}} \theta^k \|_2^2 + \lambda \| \theta^k \|_1 \right) \tag{7}$$

where θ^k denotes the sparse coefficients of y^k with respect to D_{HF}, and λ is a regularization parameter. In this paper, we apply an efficient online dictionary learning algorithm proposed in [33] to solve (7) to obtain D_{HF}, as illustrated in Fig. 5.

Dictionary Partition and Identification: We find that the atoms constituting D_{HF} can be roughly divided into two clusters (subdictionaries) for representing the geometric and rain components of I_{HF}. Intuitively, the most significant feature for a rain atom can be extracted via "image gradient." In the proposed method, we utilize the HOG descriptor [15] to describe each atom in D_{HF}. The HOG method [15] is briefly described as follows.

The basic idea of HOG is that local object appearance and shape can be usually well characterized by the distribution of local intensity gradients or edge directions, without precisely knowing the corresponding gradient or edge positions [15]. To

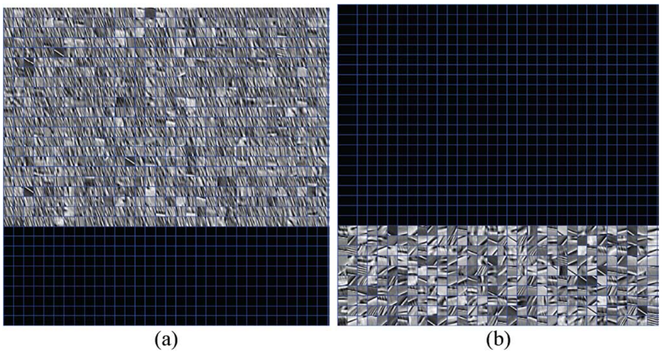


Fig. 6. Dictionary partition for the dictionary shown in Fig. 5: (a) rain subdictionary; and (b) geometric or nonrain subdictionary.

extract the HOG feature from an image, the image can be divided into several small spatial regions or cells. For each cell, a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell can be accumulated. The combined histogram entries of all cells form the HOG representation of the image. In our implementation, the size of a local image patch/dictionary atom is chosen to be 16 × 16, which leads to reasonable computational cost in dictionary partition (involving HOG feature extraction), as will be shown in Section IV.

After extracting the HOG feature for each atom in D_{HF}, we then apply the K-means algorithm to classify all of the atoms in D_{HF} into two clusters D_1 and D_2 based on their HOG feature descriptors. The following procedure is to identify which cluster consists of rain atoms and which cluster consists of geometric or nonrain atoms. First, we calculate the variance of gradient direction for each atom d_{ij}, $j = 1, 2, \ldots, N$, in cluster D_i as VG_{ij}, where N_i denotes the number of atoms in D_i, $i = 1, 2$. Then, we calculate the mean of VG_{ij} for each cluster D_i as MVG_i. Based on the fact that the edge directions of rain streaks in an atom are usually consistent, i.e., the variance of gradient direction for a rain atom should be small, we identify the cluster with the smaller MVG_i as rain subdictionary $D_{\mathrm{HF\_}R}$ and the other one as geometric (or nonrain) subdictionary $D_{\mathrm{HF\_}G}$, as depicted in Fig. 6.

On the other hand, although the dictionary learning step in the proposed method can be fully self-contained, where no extra training samples are required, the decomposition performance can be further improved by collecting a set of exemplar patches from the HF parts of some training nonrain images to learn an extended dictionary D_E to enrich the dictionary. Fig. 7 illustrates an example of D_E. Then, we integrate D_E with $D_{\mathrm{HF\_}G}$ of each image to form the final geometric subdictionary of the image.

Moreover, based on our experiences, it is hard to learn a rain dictionary by collecting a set of real exemplar rain patches due to the following reasons: 1) it is not easy to collect pure rain patches extracted from natural rain images because rain streaks are usually highly mixed with the nonrain part in an image; 2) it is also not easy to learn a dictionary adapted to a wide range of lighting and viewing conditions for rain streaks; and 3) although a few photorealistic rendering techniques have been proposed [5], it is still not easy to synthesize all possible real rain patches for learning a representative rain dictionary adapted to a large variety of rain streaks. Hence, we proposed a self-learning approach that learns the rain dictionary for a rain image based on a set of exemplar patches extracted from the image itself, followed by performing dictionary partition. The rain dictionary learned from an image itself is more appropriate to sparsely represent the rain component of the image.

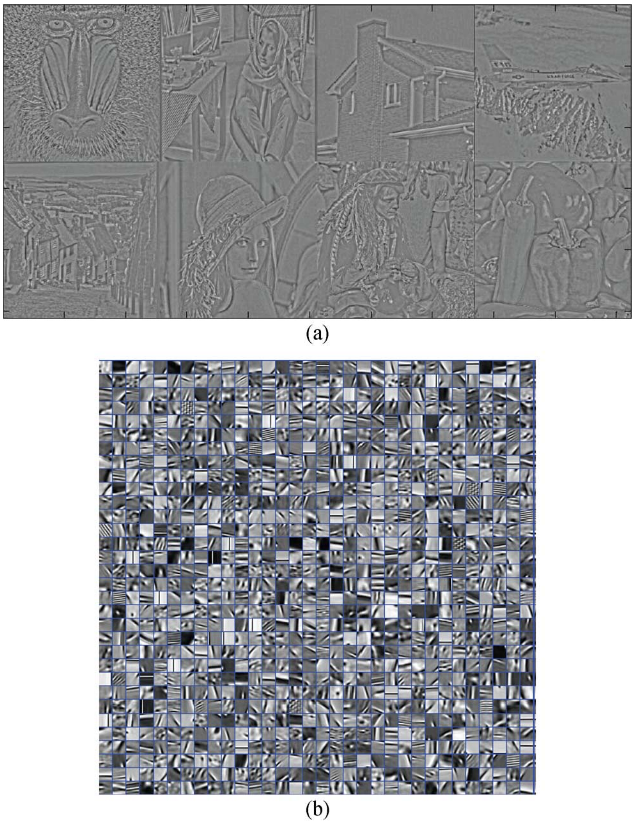


Fig. 7. Extended dictionary D_E learning: (a) the HF parts of the eight training nonrain images; and (b) learned extended dictionary.

Diversities of Two Subdictionaries: The MCA algorithms distinguish between the morphological components by taking advantage of the diversities of two dictionaries D_1 and D_2, which can be measured by the mutual incoherence of them [28]. The mutual coherence $\mu(D_1, D_2)$ between D_1 and D_2 can be defined as

$$\mu(D_1, D_2) = \max_{d_{1i} \in D_1, d_{2j} \in D_2} |\langle d_{1i}, d_{2j} \rangle| \tag{8}$$

where d_{1i} and d_{2j} stand for the ith and jth atoms (rearranged as a column vector) in $D_1$and D_2, respectively, and $\langle d_{1i}, d_{2j} \rangle$ denotes the inner product of d_{1i} and d_{2j}. When each atom is normalized to have a unit l_2 norm, the range of $\mu(D_1, D_2)$ is [0, 1]. As a result, the mutual incoherence is $1 - \mu(D_1, D_2)$. The smaller the mutual coherence is, the larger the diversities of the two subdictionaries will be and, thus, the better the decomposition performance based on the two dictionaries will be. The experimental evaluations of the mutual incoherence of rain subdictionary $D_{\mathrm{HF\_}R}$ and geometric subdictionary $D_{\mathrm{HF\_}G}$ for decomposing a rain image in the proposed method are presented in Section IV.

TABLE II
PROPOSED SINGLE-IMAGE-BASED RAIN STREAK REMOVAL ALGORITHM

Input: a rain image I.
Output: the rain-removed version I^{Non_Rain} of I.

1. Apply the bilateral filter to obtain the LF part I_{LF} and HF part I_{HF}, such that $I = I_{LF} + I_{HF}$.
2. Extract a set of image patches $y^k \in R^n, k = 1,2,\dots,P$, from I_{HF}. Apply the online dictionary learning for sparse coding algorithm to solve
$$\min_{D_{HF}\in R^{n\times m},\theta^k\in R^m} \frac{1}{P}\sum_{k=1}^{P}\left(\frac{1}{2}\|y^k - D_{HF}\theta^k\|_2^2 + \lambda\|\theta^k\|_1\right)$$
to obtain the dictionary D_{HF} consisting of the atoms that can sparsely represent $y^k, k = 1,2,\dots,P$.
3. Extract HOG feature descriptor for each atom in D_{HF}. Apply K-means algorithm to classify all of the atoms into two clusters based on their HOG feature descriptors.
4. Identify one of the two clusters as "rain sub-dictionary," D_{HF_R} and the other one as "geometric sub-dictionary," D_{HF_G}.
5. Apply MCA by performing OMP to solve
$$\min_{\theta_{HF}^k\in R^m}\|b_{HF}^k - D_{HF}\theta_{HF}^k\|_2^2 \ s.t. \|\theta_{HF}^k\|_0 \le L$$
for each patch $b_{HF}^k \in \mathrm{R}^n$, $k = 1, 2, \dots, P$, in I_{HF} with respect to $D_{HF} = [D_{HF\_G}|D_{HF\_R}]$.
6. Reconstruct each patch b_{HF}^k to recover either geometric component I_{HF}^G or rain component I_{HF}^R of I_{HF} based on the corresponding sparse coefficients obtained from Step 5.
7. Return the rain-removed version of I via $I^{Non\_Rain} = I_{LF} + I_{HF}^G$.

D. Removal of Rain Streaks

Based on the two dictionaries D_{HF_R} and D_{HF_G}, we perform sparse coding by applying the OMP algorithm [32] for each patch b_{HF}^k extracted from I_{HF} via minimization of (5) to find its sparse coefficients $\tilde{\theta}_{\mathrm{HF}}^k$. Different from traditional MCA algorithms, where sparse coding and dictionary learning should be iteratively performed, we perform sparse coding only once for each patch b_{HF}^k with respect to $D_{\mathrm{HF}} = [D_{\mathrm{HF}\_G}|D_{\mathrm{HF}\_R}]$.

Then, each reconstructed patch b_{HF}^k can be used to recover either geometric component I_{HF}^G or rain component I_{HF}^R of I_{HF} based on the sparse coefficients $\tilde{\theta}_{\mathrm{HF}}^k$ as follows. We set the coefficients corresponding to $D_{\mathrm{HF}\_\mathrm{G}}$ in $\tilde{\theta}_{\mathrm{HF}}^k$ to zeros to obtain $\tilde{\theta}_{\mathrm{HF}\_R}^k$, whereas the coefficients corresponding to D_{HF_R} in $\tilde{\theta}_{\mathrm{HF}}^k$ to zeros to obtain $\tilde{\theta}_{\mathrm{HF}\_G}^k$. Therefore, each patch b_{HF}^k can be re-expressed as either $\tilde{b}_{\mathrm{HF}\_G}^k = D_{\mathrm{HF}\_G} \times \tilde{\theta}_{\mathrm{HF}\_G}^k$ or $\tilde{b}_{\mathrm{HF}\_R}^k = D_{\mathrm{HF}\_R} \times \tilde{\theta}_{\mathrm{HF}\_R}^k$, which can be used to recover I_{HF}^G or I_{HF}^R, respectively, by averaging the pixel values in overlapping regions. Finally, the rain-removed version of the image I can be obtained via $I^{\mathrm{Non\_Rain}} = I_{\mathrm{LF}} + I_{\mathrm{HF}}^G$, as illustrated in Fig. 4(e). In summary, the proposed single-image-based rain streak removal method is summarized in Table II.

IV. EXPERIMENTS AND DISCUSSION

A. Performance Evaluation

Because we cannot find any other single-frame-based approach, to evaluate the performance of the proposed algorithm, we first compare the proposed method with a low-pass filtering method called the bilateral filter proposed in [34], which has been extensively applied and investigated recently for image processing such as image denoising [35]. In addition, to demonstrate that existing image denoising methods cannot well address the problem of single-image-based rain removal, we also compare the proposed method with the state-of-the-art image denoising method based on K-SVD dictionary learning and sparse representation proposed in [20] with a released source code available from [23] (denoted by "K-SVD-based denoising").

To the best of our knowledge, no standard still rain image data set is currently available for benchmarking. Existing video-based rain removal approaches [1]–[3], [6], [7] were all evaluated by collecting a few video frames filmed by the authors or extracted from existing movie files. Hence, we collected several natural/synthetic rain images from the Internet and also from the photorealistically rendered rain video frames (with ground-truth images) provided in [5] for a few of them. For natural rain images, it is not easy to provide many quantitative analyses due to the fact that the ground-truth images are usually unavailable. In current video-based rain removal research [2], [3], [6], [7], the performances were usually subjectively evaluated. On the other hand, to evaluate the quality of a rain-removed image with a ground-truth image, we used the visual information fidelity (VIF) metric [39] in the range of [0, 1], which has been shown to outperform peak signal-to-noise ratio metric. More test results can be found in our project website [42], where our test image data set can be downloaded.

The synthesized rain images shown in Figs. 1(b) [40] and 8(b) [41] were generated by adding rain streaks to Figs. 1(a) and 8(a), respectively, using the Photoshop software [40], [41]. On the other hand, the rendered rain images shown in Figs. 9–11 were generated by a photorealistic rendering technique proposed in [5], briefly described as follows. In [5], a rain streak appearance model capturing the complex interactions between the lighting direction, viewing direction, and oscillating shape of a raindrop was proposed. This model is built upon a raindrop oscillation model, which was developed in atmospheric sciences. The oscillation parameters were empirically decided by measuring rain streak appearances under a wide range of lighting and viewing conditions. Based on these parameters, rain streaks with variations in streak appearance with respect to lighting and viewing directions were rendered. An efficient image-based rendering algorithm was also proposed in [5] to add rain to an image or video, which requires a coarse depth map of the scene, and the locations and properties of the light sources. It should be noted that the proposed method does not assume that any knowledge about the rain streak appearance in a rain image can be available in advance.

In addition, we also compare our method with a video-based rain removal method based on adjusting camera parameters proposed in [3] (denoted by "video-based camera see"), which should outperform most of other video-based techniques without adjusting cameras. We captured some single frames from the videos released from [3] and compared our results with the ones of [3] from the same videos. For each video released in [3], the preceding frames are rain frames, followed by succeeding rain-removed frames in the same scene. We pick a single rain frame from the preceding frames for rain removal and compared our results with the rain-removed one [3] of a

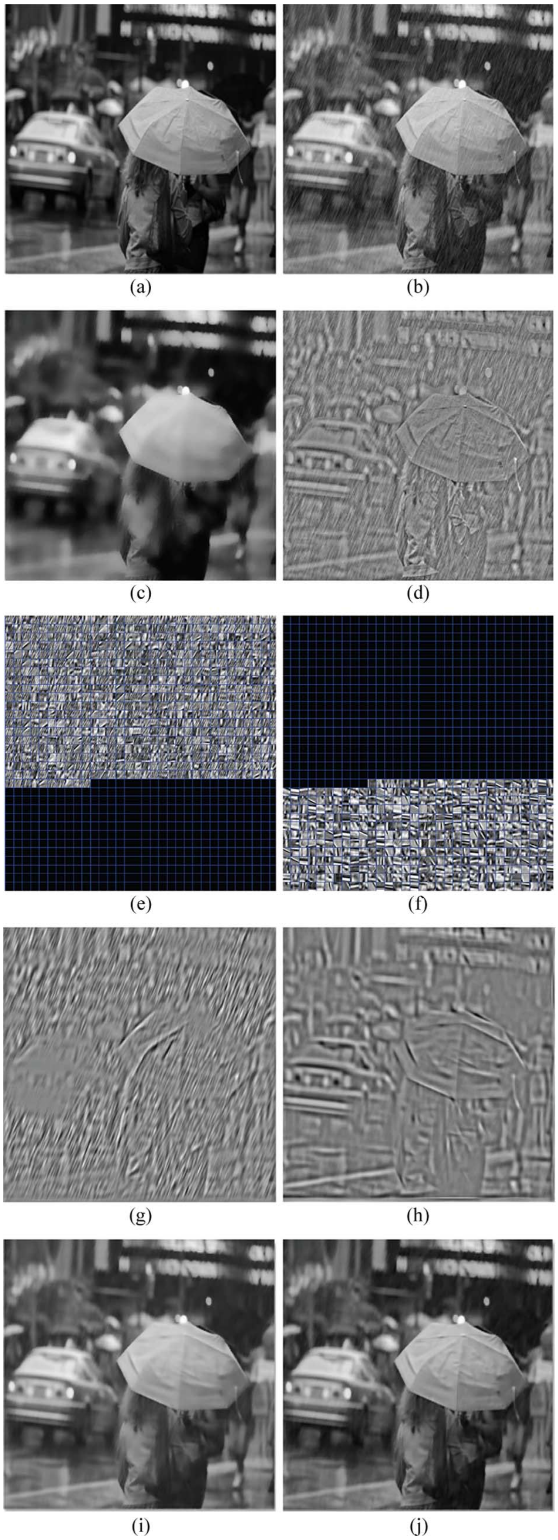


Fig. 8. Rain removal results: (a) original nonrain image (ground-truth); (b) the rain image of (a); (c) the rain-removed version of (b) via the bilateral filter $(\mathrm{VIF} = 0.31)$; (d) the HF part of (b); (e) the rain subdictionary for (d); (f) the geometric subdictionary for (d); (g) the rain component of (d); (h) the geometric component of (d); (i) the rain-removed version of (b) via the proposed method $(\mathrm{VIF} = 0.53,\ \mu = 0.7618)$; (j) the rain-removed version of (b) via the proposed method with D_E $(\mathrm{VIF} = 0.57)$.

similar frame from the succeeding frames in the same video (no exactly the same frame is available for comparison).

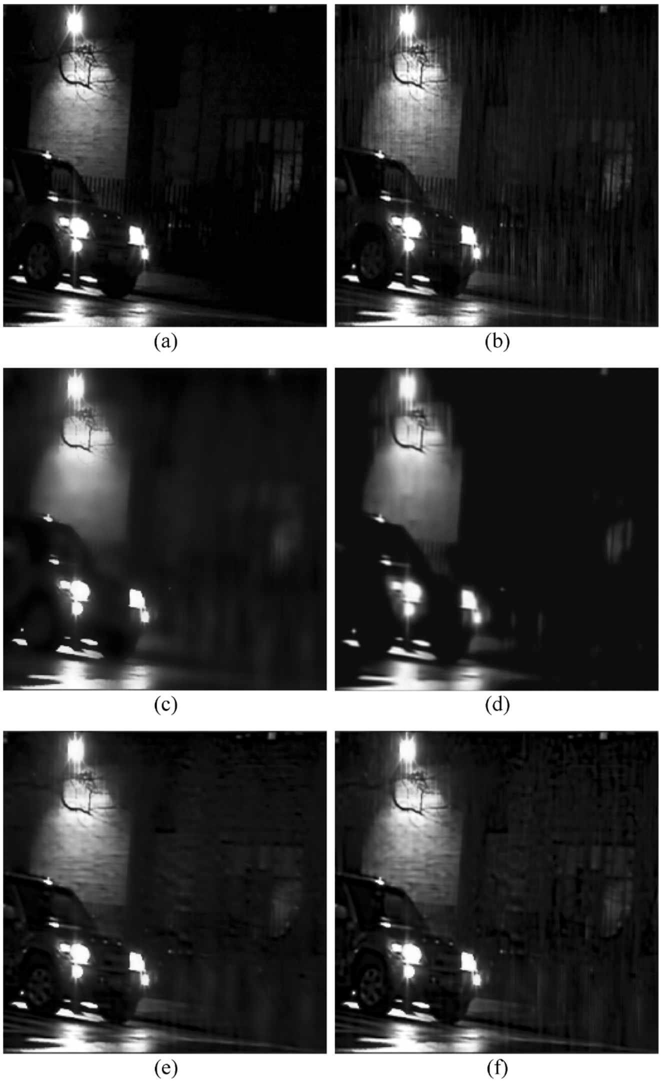


Fig. 9. Rain removal results: (a) original nonrain image; (b) the rain image of (a); the rain-removed versions of (b) via the (c) bilateral filter $(\mathrm{VIF} = 0.21)$; (d) K-SVD-based denoising method [20] $(\mathrm{VIF} = 0.21)$; (e) proposed method $(\mathrm{VIF} = 0.36,\ \mu = 0.7467)$; and (f) proposed method with D_E $(\mathrm{VIF} = 0.38)$.

The parameter settings of the proposed method are described as follows. The implementation of the bilateral filter is provided by [43], where we set the spatial-domain and intensity-domain standard deviations to 6 and 0.2, respectively, to ensure that most rain streaks in a rain image can be removed. In the dictionary learning step, we used an efficient implementation provided in [33] with the suggested regularization parameter λ used in (7) set to 0.15, which is also suggested by the sparse coding process performed in [24]. It should be noted that parameters λ's used in (2), (3), (4), (6), and (7) have the same meaning, and hence, we used the same symbol λ for convenience. In fact, only (7) is used in the proposed method. For each test grayscale image of size $N_1 \times N_2$ $(= 256 \times 256$ in our experiments), the patch size, number of training patches, dictionary size, and the number of training iterations are set to $N = 16 \times 16, P = (N_1 - \sqrt{n} + 1) \times (N_2 - \sqrt{n} + 1), m = 1024$, and 100, respectively. We also used the efficient OMP implementation provided in [33] with the number of nonzero coefficients set to at most 10, as suggested in [33]. That is, L in (5) is

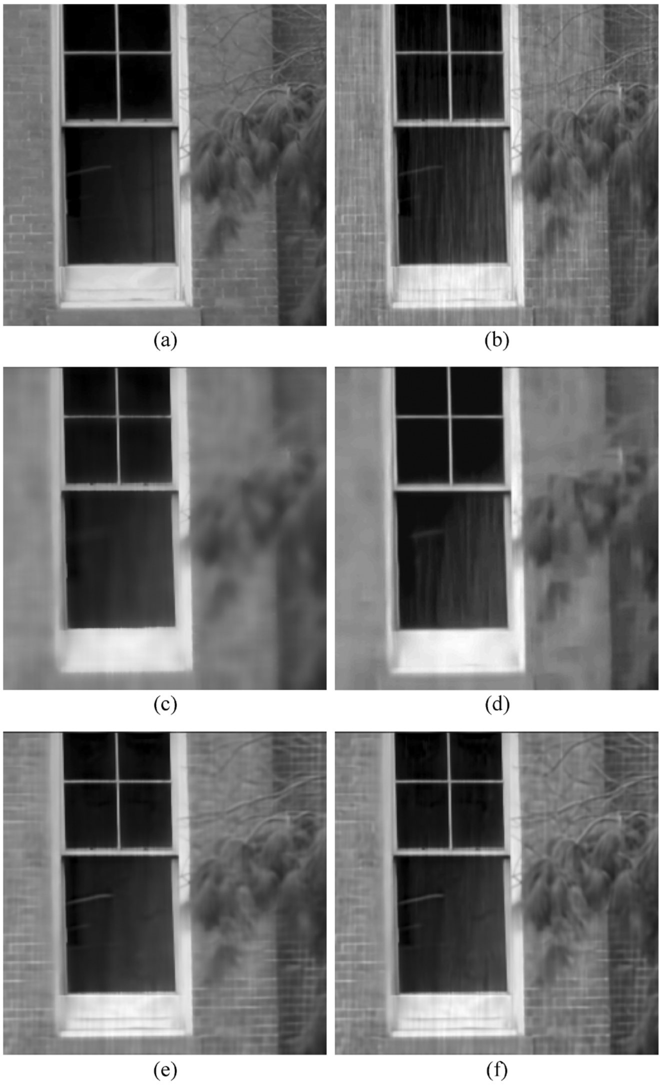


Fig. 10. Rain removal results: (a) original nonrain image; (b) the rain image of (a); the rain-removed versions of (b) via the (c) bilateral filter ($\mathrm{VIF} = 0.29$), (d) K-SVD-based denoising method ($\mathrm{VIF} = 0.38$), (e) proposed method ($\mathrm{VIF} = 0.56,\ \mu = 0.8081$), and (f) proposed method with D_E ($\mathrm{VIF} = 0.60$).

set to 10. The smaller the value of L is, the sparser the solution of (5) becomes, and vice versa. A smaller value of L leads to lower computational complexity but fewer atoms in the dictionary, and vice versa. We evaluated several possible values of L and found that $L = 10$ achieves the best tradeoff in most test cases. The used HOG implementation is provided by [16] with the dimension of each feature descriptor set to 81. The number of iterations for K-means clustering is 100.

We also evaluate the performance of the proposed method with extended dictionary D_E that is integrated with the respective geometric subdictionary for each test image. We collected several training patches extracted from the HF parts of eight widely used nonrain images, including *Baboon*, *Barbara*, *F-16*, *Goldhill*, *House*, *Lena*, *Man*, and *Pepper* images. The patch size, dictionary size, and number of training iterations are set to 16 × 16, 1024, and 200, respectively. The D_E learning process is offline performed only once. The eight training images and D_E are shown in Fig. 7.

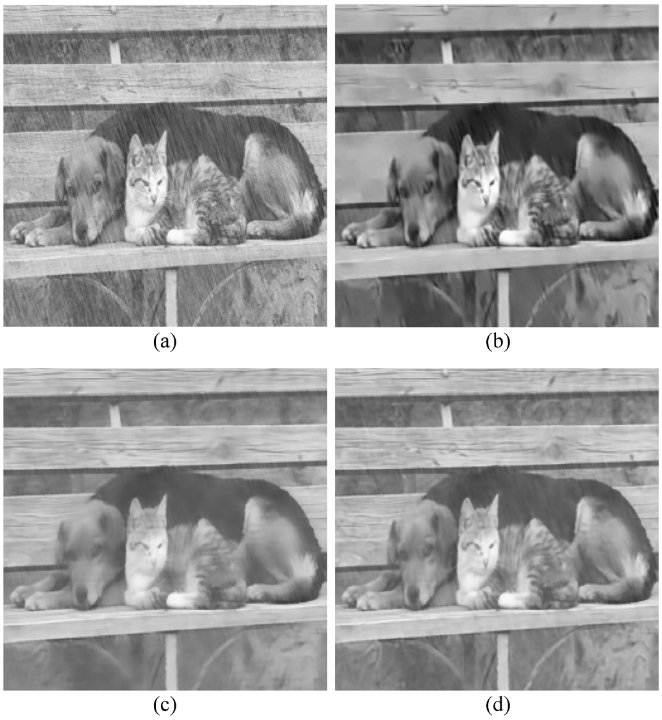


Fig. 11. Rain removal results: (a) original rain image; and the rain-removed versions of (a) via the (b) K-SVD-based denoising method, (c) proposed method ($\mu = 0.7593$), and (d) proposed method with D_E.

In addition, we compare our method with the K-SVD-based image denoising method [20], in which only one dictionary is used for the sparse coding stage, based on the assumption that the standard deviation of noise, which is assumed to be Gaussian distributed, can be known in advance. We set the parameters used in the K-SVD method according to the suggestions in [20] and [23], where the patch size, dictionary size, and the number of training iterations are set to 8 × 8, 256, and 15, respectively. In rain removal applications, the standard deviation value of the rain noise is usually unknown. To estimate the standard deviation of the rain noise, for a rendered rain image with a ground truth, we direct calculate the deviation of each rain component patch as an initial value, whereas for a natural rain image, we use the rain component obtained from the proposed method to estimate the initial value. We then manually tune the value around the initial value to ensure that most of the rain streaks in the rain image can be removed.

The rain removal results obtained from the bilateral filter [34], the K-SVD method [20], and the proposed method with and without D_E are shown in Figs. 9–13, where the test images in Figs. 9–11 are rendered rain images provided in [5]. The VIF results for the test images are summarized in Table III. The simulation results demonstrate that, although the bilateral filter and the K-SVD-based denoising filter [20] can remove most rain streaks, they both simultaneously remove much image detail as well. The proposed methods successfully remove most rain streaks while preserving most nonrain image details in these test cases, thereby improving the subjective visual quality significantly. It can be observed that the K-SVD usually cannot do a good job in rain streaks removal due to the following two

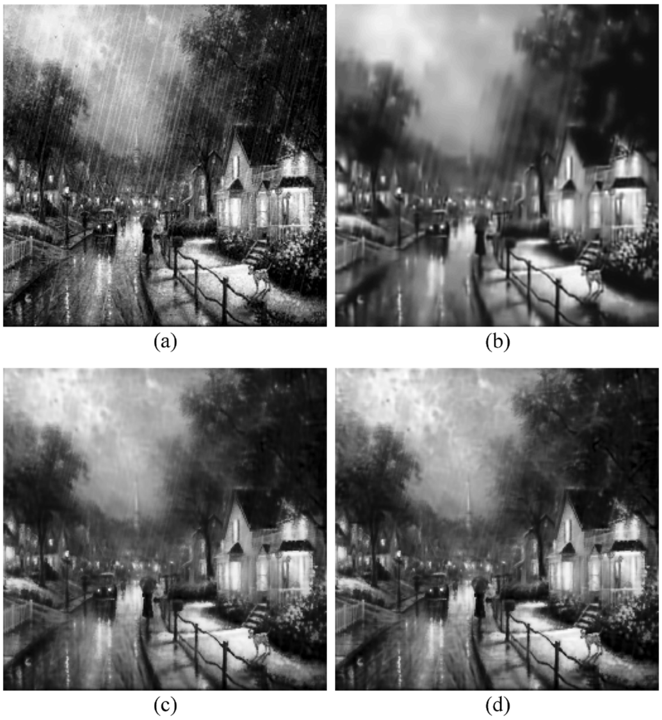


Fig. 12. Rain removal results: (a) original rain image; and the rain-removed versions of (a) via the (b) K-SVD-based denoising method, (c) proposed method ($\mu = 0.6758$), and (d) proposed method with D_E.

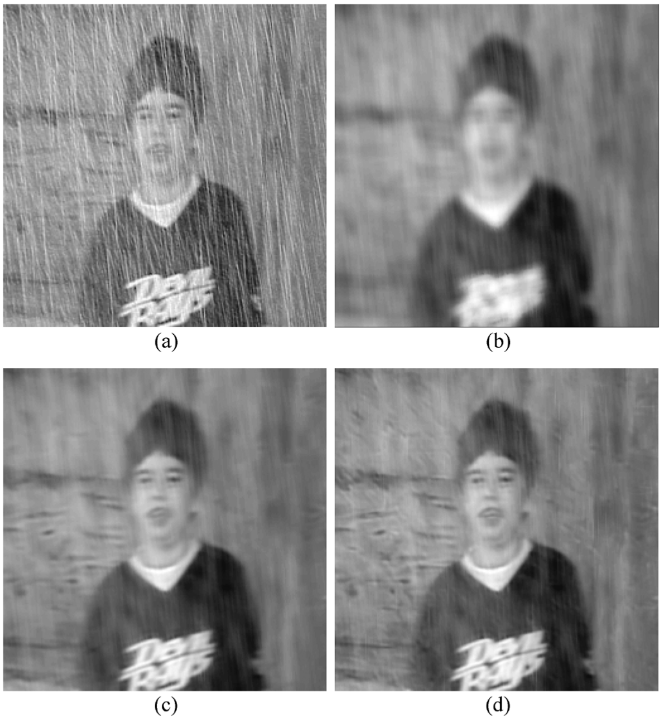


Fig. 13. Rain removal results: (a) original rain image; and the rain-removed versions of (a) via the (b) bilateral, (c) proposed method, and (e) proposed method with D_E.

TABLE III
PERFORMANCE (IN VIF VALUE) COMPARISONS AMONG THE BILATERAL FILTER [34], THE K-SVD-BASED DENOISING METHOD [20], THE PROPOSED METHOD, AND THE PROPOSED METHOD WITH D_E

	Bilateral filter	K-SVD-based denoising	Proposed	Proposed with D_E
Fig. 4	0.33	0.34	0.50	**0.52**
Fig. 8	0.31	0.41	0.53	**0.57**
Fig. 9	0.21	0.21	0.36	**0.38**
Fig. 10	0.29	0.38	0.56	**0.60**

reasons. First, the dictionary learned from nonrain natural images usually contains atoms that can represent rain steaks sufficiently well as rain streaks have similar content with many of nonrain image details. As a result, the rain part would not be neglected by the dictionary when the sparsity constraint is imposed since there will be quite a few nonzero coefficients corresponding to these rainlike atoms, particularly when the nonrain component is highly mixed with the rain streaks. Zeroing out smaller nonzero coefficients will remove both the rain part and the details of the nonrain part, thereby resulting in a seriously blurred de-rained image, as can be observed from the test results. The second reason is that the K-SVD scheme assumes that some knowledge about the noise statistics is known in advance (e.g., the standard deviation of the noise for denoising [20], [23] or even the additional information about the location of the noise for overlay text removal or inpainting [23], [24]) so that the reconstruction error can be well bounded. However, the assumption is not valid for rain streaks removal applications, as it is difficult to obtain or estimate the information in such applications.

In contrast, considering that rain streaks are usually coherent in an image and still significantly differ from the geometrical component in most parts of the image, our method addresses the aforementioned problems by using two separate dictionaries that learns the rain atoms and nonrain atoms from the input image itself for sparsely representing the rain and nonrain components of the image. Our approach further makes use of the fact that the input image contains the rain patches that are the best for training the rain atoms and the gradient features of rain patches in an image has similar statistics in terms of gradient magnitude and directions (e.g., the HOG features).

Moreover, the results obtained from the "video-based camera see" method [3] and proposed methods are compared in Fig. 14 (more results can be found in [42]). The simulation results demonstrate that, when rain streaks are obviously visible in a single frame, the proposed method achieves comparable visual quality with existing video-based methods without the need of using temporal information of successive frames and adjusting camera parameters.

It can be observed in Figs. 4 and Figs. 8.–14 that, as compared to the proposed methods without and with extended dictionary D_E, incorporating the extended geometric dictionary leads to better visual quality while increasing the computational complexity (see the run-time analysis in Table IV shown below) of sparse coding due to the much larger size of the extended dictionary. The reason why sparse coding with an extended geometric dictionary usually achieves better visual quality than that

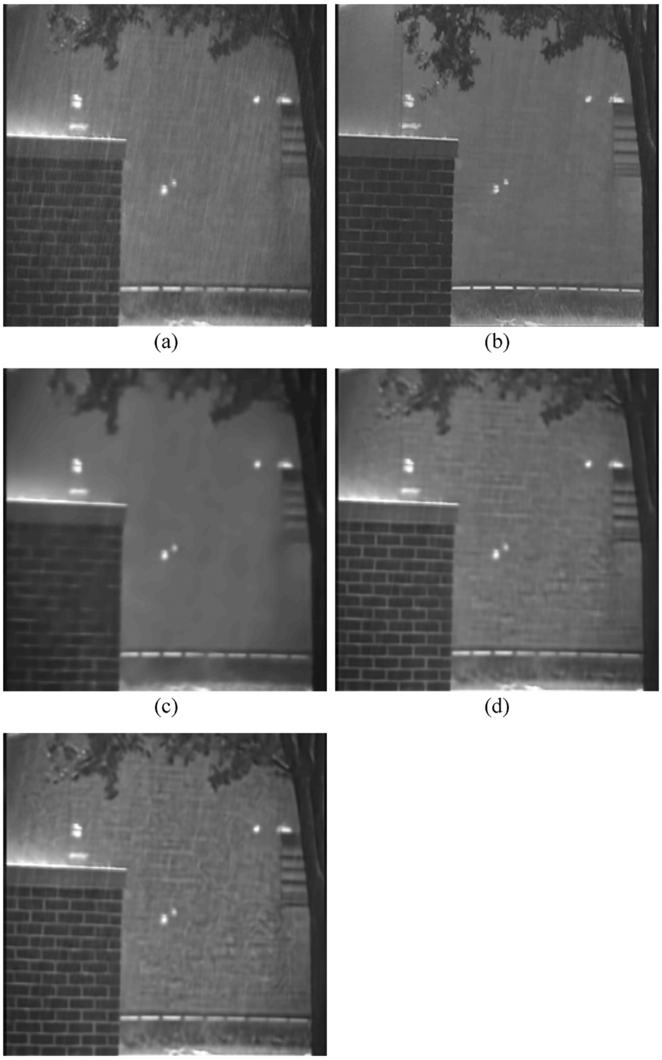

(a) (b) (c) (d) (e)

Fig. 14. Rain removal results: (a) original rain image; and the rain-removed versions of (a) via the (b) video-based camera see method [3], (c) K-SVD-based denoising method, (d) proposed method ($\mu = 0.8740$), and (e) proposed method with D_E.

without extended dictionary is that the extended dictionary provides more nonrain atoms for sparse coding to recover rain-removed version with more image details. Note that incorporating D_E in rain removal is only an option, which leads to visual quality improvement with increased complexity in most test cases. In some rare cases, however, extended dictionary D_E may not improve the visual quality of some rain patches because using D_E can possibly derive inharmonious textures in a rain-removed image. The main reason is that D_E is a much richer dictionary learned by several nonrain image patches and can be used to speculatively recover some texture information behind the rain streaks in the rain image while applying the MCA image decomposition. Note that the values of mutual coherence μ between the two subdictionaries usually fall in the range of [0.6, 0.9], which is not very close to zero. The main reason is that the two subdictionaries used in the proposed method are generated from a single learned dictionary based on a single-feature (HOG)-based clustering. It is unavoidable that the two dictionaries may have few somewhat coherent atoms, which will dominate the μ value. In the literature reporting μ values, minimization of the μ between a sensing matrix and a fixed dictionary for learning an optimal sensing matrix was mentioned in [44]. In [44], some t-averaged μ values (approaching μ when t grows) between two matrices were reported to be in a range of [0.4, 0.6], where one matrix is randomly initialized. Hence, based on the obtained rain removal results of our method and the comparison of the ranges of μ between our method and [44], the μ values of our method are usually small enough.

TABLE IV
RUN TIME (IN SECONDS) AND MEMORY USAGE ANALYSIS OF KEY OPERATIONS IN THE PROPOSED METHOD

	Bilateral filtering	Dictionary learning	Dictionary partition	Sparse coding		Total time	
				w/o	with	w/o	with
Fig. 8	1.51	72.99	2.18	8.58	24.07	87.51	101.61
Fig. 9	1.48	66.30	2.17	7.84	22.82	80.21	94.39
Fig. 10	1.50	66.21	2.28	7.00	21.45	79.17	94.06

The memory usage of the proposed method: 256K bytes (without extended dictionary) and 512K bytes (with extended dictionary).

The proposed method was implemented in MATLAB on a personal computer equipped with Intel Core i5-460M processor and 4-GB memory. The run time of each key step, including the bilateral filtering, dictionary learning, dictionary partition, and sparse coding (without and with D_E), for each test image (see Figs. 8.–12) is listed in Table IV. It can be found that the run time of the dictionary learning step dominates the total run time, which may be further reduced for future work. In Table IV, we also indicate the memory usage of our method, which is mainly dominated by the memory used for the sparse coding dictionary. In our method without extended dictionary, the self-learned dictionary contains a total of 1024 atoms, where each atom consumes 16 × 16 bytes, leading to a memory usage of 256 KB. In addition, the extended dictionary consumes additional 1024 atoms, thereby requiring 512 KB in total if the extended dictionary is utilized.

B. Discussion

There are some possible ways to further improve the visual qualities of rain-removed images. In addition to collecting training exemplar patches from some training nonrain images for learning D_E, we may extract training patches from the same or neighboring camera(s) when extending the proposed method to video rain removal. That is, we may extract exemplar patches from the neighboring rain-removed frames captured by intra/inter cameras in the same scene. Then, we can integrate the geometric subdictionary obtained from the HF part itself and the extended dictionary learned from the precollected training patches to form the final geometric subdictionary.

Moreover, the shared-private factorization scheme proposed in [45] may be used to further improve the performance of image decomposition. In [45], to best leverage the information contained in each view of an image represented by multiple views/ modalities, inspired by structured spare coding, the authors proposed an approach to learning factorized representations of multiview data in which the information is correctly factorized into components that are shared across several views and private to each view. The concept of shared-private factorization [45] may be applied to further improve our work in two aspects. First, a rain image can be segmented into several local regions with different local characteristics, which can be viewed as the multiple views. We may apply the multiview learning to learn a latent space that can separate the information (rain atoms) shared among several views from the information (unique nonrain atoms) private to each view. Then, the two dictionaries for rain removal can be accordingly identified. Second, rather than learning two disjoint private dictionaries without any common atoms for the rain and nonrain components, the two dictionaries may share some common atoms (i.e., the shared dictionary). Soft clustering (e.g., the fuzzy C-means) rather than hard clustering, or shared-private factorization, can be used to obtain better sparse representations.

V. Conclusion and Future Work

In this paper, we have proposed a single-image-based rain streak removal framework by formulating rain removal as an MCA-based image decomposition problem solved by performing sparse coding and dictionary learning algorithms. The dictionary learning of the proposed method is fully automatic and self-contained where no extra training samples are required in the dictionary learning stage. We have also provided an optional scheme to further enhance the performance of rain removal by introducing an extended dictionary of nonrain atoms learned from nonrain training images. Our experimental results show that the proposed method achieves comparable performance with state-of-the-art video-based rain removal algorithms without the need of using temporal or motion information"" for rain streak detection and filtering among successive frames.

For future work, the performance of our method may be further improved in terms of computational complexity and visual quality by enhancing the sparse coding, dictionary learning, and dictionary partitioning processes. More specifically, since the dictionary learning and sparse coding consume most of execution time, the input image can be segmented into several local regions with different local characteristics such that the online dictionary learning for individual regions can be performed in parallel to accelerate the two processes by taking advantage of current multicore processor technology. In addition, with the localized learning and sparse coding, the number of patches of each local region for dictionary learning and the number of atoms for sparse coding will be significantly fewer than those for the whole-image-based learning, thereby further reducing the computational complexities of the two processes. Nevertheless, the impact of localized learning and sparse coding on rain removal performance needs in-depth investigation. Moreover, the dictionary learning process can be further improved to obtain more accurate sparse representations by taking into account the information shared in rain and nonrain components, and the information shared in the rain components of different local regions, as mentioned in Section IV.

References

[1] P. C. Barnum, S. Narasimhan, and T. Kanade, "Analysis of rain and snow in frequency space," *Int. J. Comput. Vis.*, vol. 86, no. 2/3, pp. 256–274, Jan. 2010.

[2] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2004, vol. 1, pp. 528–535.

[3] K. Garg and S. K. Nayar, "When does a camera see rain?," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2005, vol. 2, pp. 1067–1074.

[4] K. Garg and S. K. Nayar, "Vision and rain," *Int. J. Comput. Vis.*, vol. 75, no. 1, pp. 3–27, Oct. 2007.

[5] K. Garg and S. K. Nayar, "Photorealistic rendering of rain streaks," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 996–1002, Jul. 2006.

[6] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng, "Rain removal in video by combining temporal and chromatic properties," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Toronto, ON, Canada, Jul. 2006, pp. 461–464.

[7] N. Brewer and N. Liu, "Using the shape characteristics of rain to identify and remove rain from video," *Lecture Notes Comput. Sci.*, vol. 5342/2008, pp. 451–458, 2008.

[8] 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. J. Comput. Vis.*, vol. 93, no. 3, pp. 348–367, Jul. 2011.

[9] M. S. Shehata, J. Cai, W. M. Badawy, T. W. Burr, M. S. Pervez, R. J. Johannesson, and A. Radmanesh, "Video-based automatic incident detection for smart roads: The outdoor environmental challenges regarding false alarms," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 2, pp. 349–360, Jun. 2008.

[10] M. Roser and A. Geiger, "Video-based raindrop detection for improved image registration," in *IEEE Int. Conf. Comput. Vis. Workshops*, Kyoto, Sep. 2009, pp. 570–577.

[11] J. C. Halimeh and M. Roser, "Raindrop detection on car windshields using geometric–photometric environment construction and intensity-based correlation," in *Proc. IEEE Intell. Veh. Symp.*, Xi'an, China, Jun. 2009, pp. 610–615.

[12] Google Goggles [Online]. Available: http://www.google.com/mobile/goggles/

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[14] H. Baya, A. Essa, T. Tuytelaarsb, and L. V. Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, Jun. 2005, vol. 1, pp. 886–893.

[16] O. Ludwig, D. Delgado, V. Goncalves, and U. Nunes, "Trainable classifier-fusion schemes: An application to pedestrian detection," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, St. Louis, MO, Oct. 2009, pp. 1–6.

[17] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, Jun. 2008, pp. 1–8.

[18] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[19] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multisc. Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.

[20] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[21] M. Aharon, M. Elad, and A. M. Bruckstein, "The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[22] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[23] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. New York: Springer-Verlag, 2010.

[24] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, to be published.

[25] Y. H. Fu, L. W. Kang, C. W. Lin, and C. T. Hsu, "Single-frame-based rain removal via image decomposition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Prague, Czech Republic, May 2011, pp. 1453–1456.

[26] J. M. Fadili, J. L. Starck, J. Bobin, and Y. Moudden, "Image decomposition and separation using sparse representations: An overview," *Proc. IEEE*, vol. 98, no. 6, pp. 983–994, Jun. 2010.

[27] J. M. Fadili, J. L. Starck, M. Elad, and D. L. Donoho, "MCALab: Reproducible research in signal and image decomposition and inpainting," *IEEE Comput. Sci. Eng.*, vol. 12, no. 1, pp. 44–63, Jan./Feb. 2010.

[28] J. Bobin, J. L. Starck, J. M. Fadili, Y. Moudden, and D. L. Donoho, "Morphological component analysis: An adaptive thresholding strategy," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2675–2681, Nov. 2007.

[29] G. Peyré, J. Fadili, and J. L. Starck, "Learning adapted dictionaries for geometry and texture separation," in *Proc. SPIE*, 2007, vol. 6701, p. 67 011T.

[30] J. L. Starck, M. Elad, and D. L. Donoho, "Image decomposition via the combination of sparse representations and a variational approach," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1570–1582, Oct. 2005.

[31] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, Jun. 1996.

[32] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[33] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Jan. 2010.

[34] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. IEEE Int. Conf. Comput. Vis.*, Bombay, India, Jan. 1998, pp. 839–846.

[35] M. Zhang and B. K. Gunturk, "Multiresolution bilateral filtering for image denoising," *IEEE Trans. Image Process.*, vol. 17, no. 12, pp. 2324–2333, Dec. 2008.

[36] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[37] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, Feb. 2009.

[38] H.-W. Chen, L.-W. Kang, and C.-S. Lu, "Dictionary learning-based distributed compressive video sensing," in *Proc. Picture Coding Symp.*, Nagoya, Japan, Dec. 2010, pp. 210–213.

[39] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 430–444, Feb. 2006.

[40] D. Tang, "Rain," Photoshop Tutorials [Online]. Available: http://photoshoptutorials.ws/photoshop-tutorials/photo-effects/rain.html

[41] S. Patterson, Photoshop Rain Effect-Adding Rain to a Photo [Online]. Available: http://www.photoshopessentials.com/photo-effects/rain/

[42] NTHU Rain Removal Project [Online]. Available: http://www.ee.nthu.edu.tw/cwlin/Rain_Removal/Rain_Removal.htm

[43] D. Lanman, Bilateral Filtering and Image Abstraction [Online]. Available: http://mesh.brown.edu/dlanman/courses.html

[44] J. M. Duarte-Carvajalino and G. Sapiro, "Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1395–1408, Jul. 2009.

[45] Y. Jia, M. Salzmann, and T. Darrell, "Factorized latent spaces with structured sparsity," in *Proc. Conf. Neural Inf. Proc. Syst.*, Vancouver, BC, Canada, Dec. 2010, pp. 982–990.

Li-Wei Kang (S'05–M'06) received the B.S., M.S., and Ph.D. degrees in computer science from National Chung Cheng University, Chiayi, Taiwan, in 1997, 1999, and 2005, respectively.

Since August 2010, he has been with the Institute of Information Science, Academia Sinica (IIS/AS), Taipei, Taiwan, as an Assistant Research Scholar. From October 2005 to July 2010, he was with IIS/AS as a Postdoctoral Research Fellow. His research interests include multimedia content analysis and multimedia communications.

Dr. Kang served as an Editorial Advisory Board Member for the book, *Visual Information Processing in Wireless Sensor Networks: Technology, Trends and Applications* (IGI Global, 2011); a Guest Editor of Special Issue on Advance in Multimedia, Journal of Computers, Taiwan; a Co-organizer of a special session on Advanced Techniques for Content-Based Image/Video Resizing of Visual Communications and Image Processing 2011 and a special session on Image/Video Processing and Analysis of APSIPA ASC 2011. He was the recipient of two paper awards from Computer Vision, Graphics, and Image Processing Conferences and Image Processing and Pattern Recognition Society, Taiwan, in 2006 and 2007, respectively.

Chia-Wen Lin (S'94–M'00–SM'04) received the Ph.D. degree in electrical engineering from National Tsing Hua University (NTHU), Hsinchu, Taiwan, in 2000.

He is currently an Associate Professor with the Department of Electrical Engineering and the Institute of Communications Engineering, NTHU. He was with the Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan, during 2000–2007. Prior to joining academia, he worked for the Information and Communications Research Laboratories, Industrial Technology Research Institute, Hsinchu, Taiwan, during 1992–2000, where his final post was a Section Manager. From April 2000 to August 2000, he was a Visiting Scholar with the Information Processing Laboratory, Department of Electrical Engineering, University of Washington, Seattle. He has authored or coauthored more than 100 technical papers. He holds more than 20 patents. His research interests include video content analysis and video networking.

Dr. Lin is an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, and the *Journal of Visual Communication and Image Representation*. He is also an Area Editor of *EURASIP Signal Processing: Image Communication*. He served as a Technical Program Cochair of the IEEE International Conference on Multimedia & Expo (ICME) in 2010, and a Special Session Cochair of the IEEE ICME in 2009. He was a recipient of the 2001 Ph.D. Thesis Awards presented by the Ministry of Education, Taiwan. His paper won the Young Investigator Award presented by International Society for Optics and Photonics Visual Communications and Image Processing 2005. He received the Young Faculty Awards presented by CCU in 2005 and the Young Investigator Awards presented by the National Science Council, Taiwan, in 2006.

Yu-Hsiang Fu received the B.S. and M.S. degrees in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2008 and 2010, respectively.

He has been with the MStar Semiconductor Inc., Hsinchu, Taiwan, as a Design Engineer since October 2010. He was an Intern with the Mechanical and Systems Research Laboratories, Industrial Technology Research Institute, in 2010. His research interests include image/video processing and computer vision.