Self-Learning Based Image Decomposition With Applications to Single Image Denoising

De-An Huang, Li-Wei Kang, *Member, IEEE*, Yu-Chiang Frank Wang, *Member, IEEE*, and Chia-Wen Lin, *Senior Member, IEEE*

***Abstract*—Decomposition of an image into multiple semantic components has been an effective research topic for various image processing applications such as image denoising, enhancement, and inpainting. In this paper, we present a novel self-learning based image decomposition framework. Based on the recent success of sparse representation, the proposed framework first learns an over-complete dictionary from the high spatial frequency parts of the input image for reconstruction purposes. We perform unsupervised clustering on the observed dictionary atoms (and their corresponding reconstructed image versions) via affinity propagation, which allows us to identify image-dependent components with similar context information. While applying the proposed method for the applications of image denoising, we are able to automatically determine the undesirable patterns (e.g., rain streaks or Gaussian noise) from the derived image components directly from the input image, so that the task of single-image denoising can be addressed. Different from prior image processing works with sparse representation, our method does not need to collect training image data in advance, nor do we assume image priors such as the relationship between input and output image dictionaries. We conduct experiments on two denoising problems: single-image denoising with Gaussian noise and rain removal. Our empirical results confirm the effectiveness and robustness of our approach, which is shown to outperform state-of-the-art image denoising algorithms.**

***Index Terms*—Denoising, image decomposition, rain removal, self-learning, sparse representation.**

I. Introduction

ASSUMING an image is a linear mixture of multiple source components, image decomposition aims at determining such components and the associated weights [1], [2]. For example, how to properly divide an image into texture and non-texture parts has been investigated in applications of image compression [3], image inpainting [4], [5], or related image analysis and synthesis tasks. Consider a fundamental problem of decomposing an image of N pixels into C different N-dimensional components, one needs to solve a linear regression problem with $N \times C$ unknown variables. While this problem is ill-posed, image sparsity prior has been exploited to address this task [1]. As a result, an input image can be morphologically decomposed into different patches based on such priors for a variety of image processing applications.

Before providing the overview and highlighting the contributions of our proposed method, we will first briefly review morphological component analysis (MCA), which is a sparse-representation based image decomposition algorithm, and has been successfully applied and extended to solve the problems of image denoising [6]–[8], image inpainting [5], [8], and image deraining (i.e., rain removal) [9], [10].

A. MCA for Image Decomposition

MCA utilizes the morphological diversity of different features contained in the data to be decomposed and to associate each morphological component to a dictionary of atoms [1], [5], [11]. Suppose an image I of N pixels is a superposition of K components (called morphological components), denoted by $I = \sum_{k=1}^{K} I_k$, where I_k denotes the k-th component, such as the geometric or textural component of the image I. To decompose I into I_k, $k = 1, 2, \ldots, K$, MCA iteratively minimizes the following energy function:

$$E(\{I_k\}_{k=1}^{K}, \{\boldsymbol{\theta}_k\}_{k=1}^{K}) = \frac{1}{2}\|I - \sum_{k=1}^{K} I_k\|_2^2 + \tau \sum_{k=1}^{K} E_k(I_k, \boldsymbol{\theta}_k), \tag{1}$$

where $\boldsymbol{\theta}_k$ denotes the sparse coefficients corresponding to I_k with respect to the dictionary $\mathbf{D}_k$, τ is a regularization parameter, and E_k is the energy function defined according to the type of $\mathbf{D}_k$ (global or local dictionary).

The MCA algorithms solve (1) by iteratively performing for each component I_k, the following two steps: (i) update of the sparse coefficients: this step performs sparse coding to solve $\boldsymbol{\theta}_k$ or $\{\boldsymbol{\theta}_k^p\}_{p=1}^{P}$, where $\boldsymbol{\theta}_k^p$ represents the sparse coefficients of the p-th patch $\mathbf{y}_k^p$ extracted from I_k, and P is the total number of extracted patches, to minimize $E_k(I_k, \boldsymbol{\theta}_k)$ while fixing I_k; and (ii) update of the components: this step updates I_k or $\{\mathbf{y}_k^p\}_{p=1}^{P}$ while fixing $\boldsymbol{\theta}_k$ or $\{\boldsymbol{\theta}_k^p\}_{p=1}^{P}$. More details about MCA can be found in [1], [5], [11].

Manuscript received December 13, 2012; revised April 18, 2013 and June 17, 2013; accepted July 08, 2013. Date of publication October 07, 2013; date of current version December 12, 2013. This paper is an extended version of the original paper which appeared in the Proceedings of IEEE ICME 2012 Conference and was among the top-rated 4% of ICME'12 submissions. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sen-Ching Cheung.

D.-A. Huang and Y.-C. F. Wang are with the Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan (e-mail: andrew800619@gmail.com; ycwang@citi.sinica.edu.tw).

L.-W. Kang is with the Graduate School of Engineering Science and Technology-Doctoral Program, and the Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Yunlin, Taiwan (e-mail: lwkang@yuntech.edu.tw).

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

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/TMM.2013.2284759

TABLE I
THE NOTATIONS AND DESCRIPTIONS OF THE SYMBOLS IN THIS PAPER

Symbols	Meanings
I, I_L, I_H	Input image to be decomposed, low-frequency part of I, and high-frequency part of I
M, K	Number of atoms, number of image components/clusters
τ, λ	Regularization parameters
$\boldsymbol{y}_H^p$	The p-th training exemplar patch of size $n \times n$ extracted from I_H
$\boldsymbol{D}_H$	Dictionary of size $n^2 \times M$ for sparsely representing I_H
$\boldsymbol{d}_i$	The i-th atom in $\boldsymbol{D}_H$
$\boldsymbol{d}_i^k$	The i-th atom of the k-th group dictionary.
$\boldsymbol{\theta}_H^p$	Sparse coefficient vector of $\boldsymbol{y}_H^p$ with respect to $\boldsymbol{D}_H$
$\boldsymbol{D}_H^k$	Dictionary for sparsely representing I_H^k
I_H^k	The k-th component of I_H

B. Overview and Contributions of the Proposed Method

In this paper, we propose a self-learning based image decomposition framework. The proposed method identifies image components based on semantical similarity and thus can be easily applied to the applications of image denoising. Unlike prior learning-based image decomposition or denoising works which require the collection of training image data (e.g., raw/noisy inputs vs. denoised outputs, or low-resolution vs. high-resolution output images), our proposed method advocates the self-learning of the input (noisy) image directly. After observing dictionary atoms with high spatial frequency (i.e., potential noisy patterns), we advance the unsupervised clustering algorithm of affinity propagation without any prior knowledge of the number of clusters, which allows us to automatically identify the dictionary atoms which correspond to undesirable noise patterns. As a result, removing such noise from the input image can be achieved by performing image reconstruction without using the associated dictionary atoms. From the above explanation, it can be seen that our proposed method does *not* require any external training image data (e.g., noisy and ground truth image pairs), and *no* user interaction or prior knowledge is needed either. Therefore, our method can be considered as an *unsupervised* approach. And, as verified by our experiments, our method can be directly applied to a single input image and solve single-image problems of rain streaks and Gaussian noise removal. The former type of noise can be considered *structured* noise patterns, and the latter as the *unstructured* ones.

The major contribution of this paper is tri-fold: (i) unlike prior MCA based approaches [1], [5], [11], our proposed method allows one to decompose an input image and to observe its representation without the need to learn from pre-collected training data. We do not assume any image priors such as the relationship between the input and desirable output images either. This makes single-image based applications applicable in real-world scenarios; (ii) we advance affinity propagation for identifying key image components which exhibit similar context information, so that those associated with noise or undesirable patterns can be disregarded for automatic image denoising; (iii) while our proposed framework can be applied to address the tasks of single-image denoising and rain removal, we further show that we do not limit our method to the use of any specific preprocessing techniques when retrieving the high spatial frequency parts (e.g., bilateral filtering [12], K-SVD-based image denoising [7], and BM3D filtering [13]). The effectiveness and robustness of our method will later be confirmed by our experiments.

The rest of this paper is organized as follows. We briefly review sparse representation and dictionary learning techniques in Section II. Section III presents our proposed framework for image decomposition via self-learning. Sections IV and V explain how to apply our method for single-image rain removal and denoising, respectively. Experimental results on two types of denoising problems will be presented in Section VI, followed by the conclusions of this paper.

II. SPARSE REPRESENTATION AND DICTIONARY LEARNING

A. Sparse Representation

Sparse coding [14]–[16] is a technique of representing a signal in terms of a compact linear combination of a set of basis signals (or atoms) from a dictionary. A pioneering work in image sparse representation [14] stated that the receptive fields of simple cells in mammalian primary visual cortex can be characterized as being spatially localized, oriented, and bandpassed. It was shown that a coding strategy that maximizes sparsity is sufficient to account for the above properties, and that a learning algorithm attempting to determine sparse linear codes for natural scenes will develop a complete family of localized, oriented, and bandpassed receptive fields.

For each image patch $\mathbf{y}^p$ extracted from an image I, we can find the corresponding sparse coefficient vector $\boldsymbol{\theta}^p$ with respect to a given dictionary $\mathbf{D}$ (described in Section II-B) by solving the following optimization problem

$$\arg\min_{\boldsymbol{\theta}^p}\left(\frac{1}{2}\|\mathbf{y}^p - \mathbf{D}\boldsymbol{\theta}^p\|_2^2 + \lambda\|\boldsymbol{\theta}^p\|_1\right), \tag{2}$$

where λ is a regularization parameter. It has been shown that (2) can be efficiently solved using the orthogonal matching pursuit (OMP) algorithm [6], [15], [17].

B. Dictionary Learning

To construct a dictionary $\mathbf{D}$ to sparsely represent each patch extracted from an image, one can use a set of training image patches $\mathbf{y}^p$, $p = 1, 2, \ldots, P$, for learning purposes. To derive a dictionary $\mathbf{D}$ which satisfies the above sparse coding scheme, we solve the following optimization problem [6], [17]:

$$\min_{\mathbf{D},\boldsymbol{\theta}^p}\sum_{p=1}^{P}\left(\frac{1}{2}\|\mathbf{y}^p - \mathbf{D}\boldsymbol{\theta}^p\|_2^2 + \lambda\|\boldsymbol{\theta}^p\|_1\right), \tag{3}$$

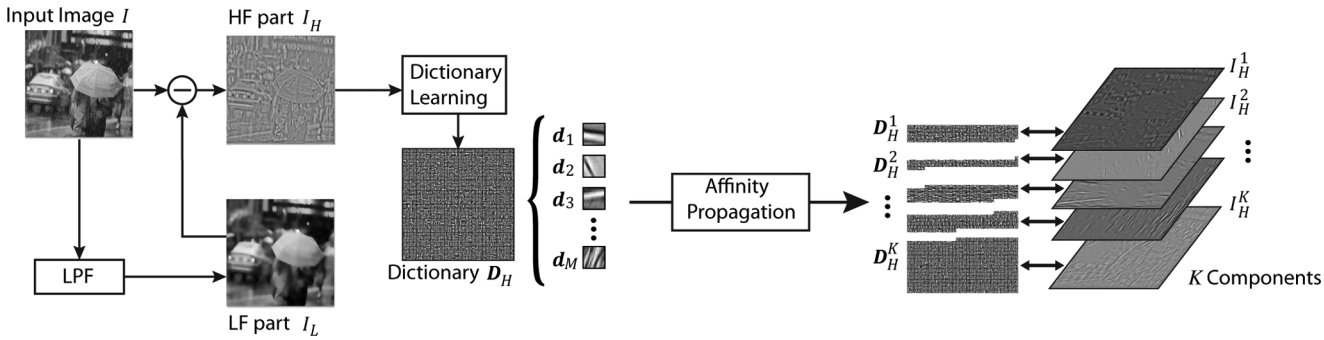

Fig. 1. Illustration of our image decomposition framework. After removing the low spatial frequency (LF) parts (i.e., I_L) of the input image I, we learn a dictionary $\mathbf{D}_H$ for representing the high spatial frequency (HF) image (i.e., I_H). The observed M dictionary atoms are grouped into K different clusters based on their context information, so that the corresponding atom set $\mathbf{D}_H^i$ is associated with a particular HF image component I_H^i for decomposition purposes.

where $\boldsymbol{\theta}^p$ denotes the sparse coefficient vector of $\mathbf{y}^p$ with respect to $\mathbf{D}$ and λ is a regularization parameter. Equation (3) can be efficiently solved by performing a dictionary learning algorithm, such as online dictionary learning [17] or K-SVD [6] algorithms.

III. Self-Learning Based Image Decomposition

Fig. 1 shows the proposed framework for image decomposition. As shown in this figure, we approach this problem by solving the task of dictionary learning for image sparse representation, followed by the learning of context-aware image components. While the former aims at reconstructing an input image using sparse representation techniques, the latter will be utilized to identify key image components based on their context information (and thus can be applied for denoising purposes). The notations used in this paper are summarized in Table I. We now detail our proposed method.

A. Dictionary Learning for Image Sparse Representation

In this paper, we focus on addressing image denoising problems. In our proposed framework, we first separate the high spatial frequency parts I_H from the low spatial frequency parts I_L for an input image I. This is because most undesirable noise patterns like rain streaks or Gaussian noise are of this type. In order to achieve $I = I_L + I_H$, we consider the use of three low-pass filtering (LPF) or denoising techniques: bilateral [12], K-SVD [7], and BM3D [13] as the preprocessing stage. We note that these LPF or denoising techniques can be replaced by band-pass filtering if the noise of interest is known to be associated with a particular frequency band. Nevertheless, we can produce I_H by subtracting the resulting smoothed/filtered version I_L from I. However, since we do not have prior knowledge or assumptions on the type of noise to be removed, it is not clear how to identify the image components of I_H which correspond to undesirable noise patterns.

As discussed in Section I, MCA has been successfully applied to decompose an image into different components/atoms. However, traditional MCA approaches usually use a fixed dictionary (e.g., discrete cosine transform (DCT), wavelet, or curvelet basis) to sparsely represent an image component. For these cases, the selection of dictionaries and parameters become heavily empirical, and the results will be sensitive to the choice of dictionaries. While some advanced training image data to learn dictionaries for improved representation, how to select a *proper* image set in advance for training remains a challenging problem. Moreover, the collection of training data might not be practical in many real-world applications such as single-image based processing tasks.

Therefore, different from the traditional MCA using fixed dictionaries, we advocate the learning of dictionary directly from the input image. More precisely, we only learn a dictionary based on the high spatial frequency part of the input image (i.e., I_H). Once such a dictionary is observed, the remaining task is to automatically identify the undesirable components/patterns which correspond to noise, so that one can perform image reconstruction without using such components for achieving image denoising.

To be more specific, we extract patches $\mathbf{y}_H^p (p = 1, 2, \ldots, P)$ of size $n \times n$ from I_H for learning the dictionary $\mathbf{D}_H$ via solving (3). We apply an online dictionary learning algorithm [17] for solving the following problem:

$$\min_{\mathbf{D}_H, \boldsymbol{\theta}_H^p} \sum_{p=1}^{P} \left(\frac{1}{2}\|\mathbf{y}_H^p - \mathbf{D}_H \boldsymbol{\theta}_H^p\|_2^2 + \lambda \|\boldsymbol{\theta}_H^p\|_1\right), \tag{4}$$

where $\boldsymbol{\theta}_H^p$ denotes the sparse coefficient vector of $\mathbf{y}_H^p$ with respect to $\mathbf{D}_H$ and λ is a regularization parameter. The learned dictionary $\mathbf{D}_H = [\mathbf{d}_1, \ldots, \mathbf{d}_M]$ contains M atoms and thus is of size $n^2 \times M$ (we have $M > n^2$).

B. Learning of Context-Aware Image Components

It is worth noting that, the atoms $\mathbf{d}_i$ of $\mathbf{D}_H$ are *not* necessarily distinct from each other in such a over-complete dictionary for sparse image representation. Therefore, it is not easy to estimate the undesirable image patterns in I_H using the observed dictionary atoms. Inspired by MCA, we proposed to separate these atoms into disjoint groups (i.e., those within the same group are semantically similar to each other). Thus, it will be possible to determine the group (and their components) associated with the noise of interest, and the task of image denoising can be achieved by performing image reconstruction without using those undesirable components.

We approach this task as solving an unsupervised clustering problem. We group the aforementioned M atoms $\mathbf{d}_m, m = 1, 2, \ldots, M$, into K different clusters, so that the atoms within the same group will share similar edge or texture information.

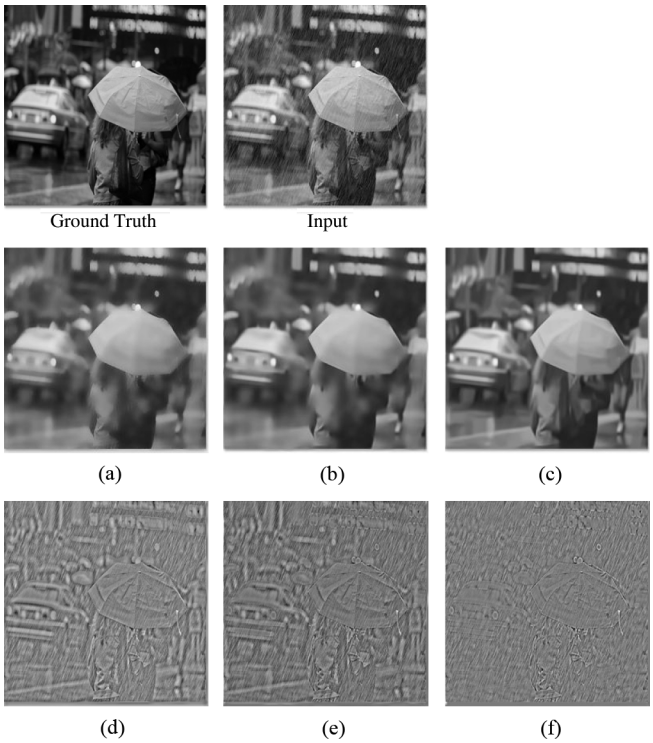

Fig. 2. Examples of deriving I_H from the input image I using different LPF techniques. The filtered/denoised versions I_L of the input using bilateral filtering, K-SVD, and BM3D are shown in (a), (b), and (c), respectively. The corresponding $I_H = I - I_L$ are shown in (d), (e), and (f), respectively.

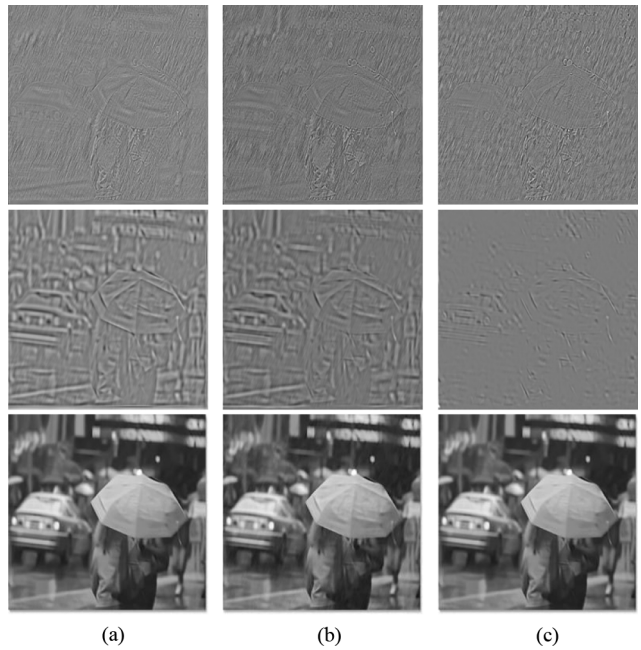

Fig. 3. Examples of rain-removal results of Fig. 2 using (a) bilateral filtering, (b) KSVD, and (c) BM3D. The first row shows the estimated rain image components. The estimated non-rain HF image components are depicted in the second row. The third row shows the final rain removal versions (i.e., integration of non-rain HF components and I_L).

Since the number of clusters K is not known, we apply affinity propagation [18] for solving this task, which minimizes the net-similarity (NS) between atoms:

$$NS = \sum_{i=1}^{M}\sum_{j=1}^{M} c_{ij}s(\mathbf{d}_i,\mathbf{d}_j) - \gamma\sum_{i=1}^{M}(1-c_{ii})(\sum_{j=1}^{M} c_{ij}) - \gamma\sum_{i=1}^{M}|(\sum_{j=1}^{M} c_{ij}) - 1|. \quad (5)$$

In (5), the function $s(\mathbf{d}_i,\mathbf{d}_j)$ measures the similarity between atoms $\mathbf{d}_i$ and $\mathbf{d}_j$. In order to group atoms share similar edge or texture information, we define the similarity function as $s(\mathbf{d}_i,\mathbf{d}_j) = exp(-\|HOG(\mathbf{d}_i) - HOG(\mathbf{d}_j)\|^2)$ where $HOG(\cdot)$ extracts the features of Histogram of Oriented Gradients [19] describing the shape/texture information of the atom. The coefficient $c_{ij} = 1$ indicates that the atom $\mathbf{d}_i$ is the exemplar (i.e., the cluster representative) of the atom $\mathbf{d}_j$, and thus $\mathbf{d}_j$ is categorized to cluster i (and c_{ii} equals 1 since $\mathbf{d}_i$ itself is the exemplar cluster i). The first term in (5) is to calculate the similarity between atoms with each cluster, while the second term penalizes the case when atoms are assigned to an empty cluster (i.e., $c_{ii} = 0$ but with $\sum_{j=1}^{M} c_{ij} \geq 1$). The third term in (5), on the other hand, penalizes the condition when atoms belong to more than one cluster, or no cluster label is assigned. In practice, the parameter γ is set to $+\infty$ to avoid the aforementioned problems. We note that, in addition to HOG, other features describing shape or textural information can also be considered in our work. For example, the features of Histogram of Orientation of Streaks (HOS) have recently been utilized in [20] for representing the patterns of rain or snow in video frames. Since our work focuses on the self-learning and decomposition of an input image, we are particularly interested in identifying and removing a dominant undesirable noise pattern from the input. From our experiments, the use of HOG features is sufficient for us to achieve this goal.

After automatically grouping the extracted M dictionary atoms into K different image clusters, we can derive image components I_H^k associated with each cluster. That is, the p-th patch of I_H^k is computed from $\mathbf{D}_H\delta_k(\boldsymbol{\theta}_H^p)$ where $\delta_k(\boldsymbol{\theta}_H^p)$ is a vector whose nonzero entries are only those associated with the atoms in the k-th cluster. Each image component I_H^k can considered as being associated to a particular type of context information, as depicted in Fig. 1. This completes the task of image decomposition. In the following sections, we will discuss how we apply this proposed framework for two denoising tasks, in which rain streaks and Gaussian noise need to be automatically identified and removed.

IV. Application to Single Image Rain Removal

A. Vision-Based Rain Removal

Since rain streaks presented in images or videos cause complex visual effects on spatial or temporal domains, which may significantly reduce user satisfaction or degrade the performances of surveillance-related applications, removal of such patterns from images/videos has recently received much attention from researchers [9], [10], [20]–[22]. Prior vision-based approaches typically focus on detecting and removing rain streaks in a video, where both the spatial and temporal information in the video can be employed for rain removal. A pioneering work on rain removal from videos was proposed in

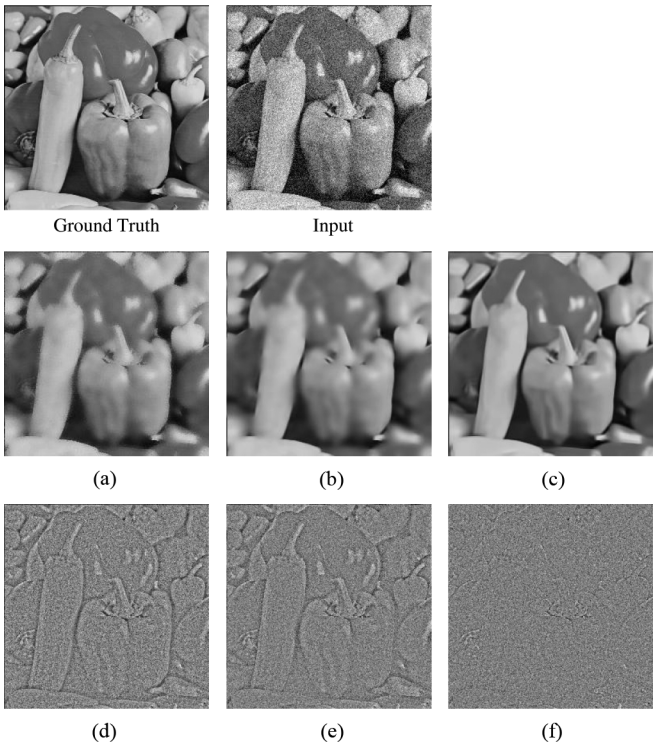

Fig. 4. Examples of deriving I_H from the input image I using different LPF techniques. The filtered/denoised versions I_L of the input using bilateral filtering, K-SVD, and BM3D are shown in (a), (b), and (c), respectively. The corresponding $I_H = I - I_L$ are shown in (d), (e), and (f), respectively.

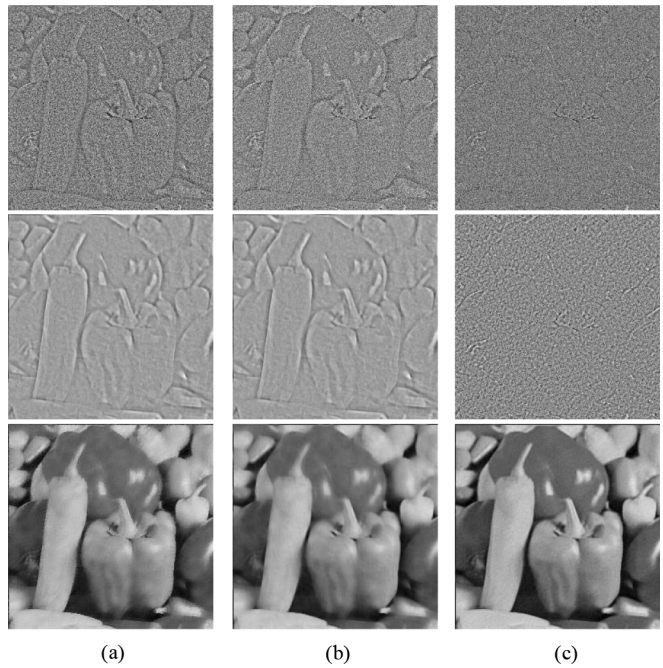

Fig. 5. Examples of denoising results of Fig. 4 using (a) bilateral filtering, (b) KSVD, and (c) BM3D. The first row shows the estimated rain image components. The estimated noise-free HF image components are depicted in the second row. The third row shows the final denoised outputs (i.e., integration of noise-free HF components and I_L).

[21], in which a correlation model capturing the dynamics of rain and a physics-based motion blur model characterizing the photometry of rain were developed.

On the other hand, when only a single image is available, it becomes a more challenging task to detect and remove such noise patterns. In [9], we have proposed a single-image rain removal framework, which approaches the rain removal task as the image decomposition problem based on MCA [1]. In this work, we first separated a rain image into low and high-frequency parts via bilateral filtering [12]. The high-frequency part was then decomposed into the "rain component" and the "non-rain component" by learning the associated sparse-representation based dictionaries for representing rain and non-rain components, respectively. To achieve this, K-means clustering (K = 2) was performed on the learned dictionary atoms for distinguishing between rain and non-rain atoms. We note that, while satisfactory results were reported in [9], their use of K-means clustering for separating the rain streak patterns from non-rain ones, and the collection of training image data for dictionary learning, might limit the performance.

B. Our Proposed Method for Single Image Rain Removal

In this study, we formulate the problem of single image rain removal as an image decomposition problem as [9] did. As illustrated in Fig. 1 and discussed in Section III, we first decompose an input rain image I into I_L and I_H using existing low-pass filtering techniques. Fig. 2 shows examples of producing I_H using different filtering techniques. Once I_H is obtained, we learn the dictionary $\mathbf{D}_H$ for representation purposes, and the dictionary atoms $\mathbf{d}_i$ will be grouped into different clusters based on its HOG features via affinity propagation. This clustering stage is to identify dictionary atoms which are similar to each other in terms of their context information. Once this stage is complete, we obtain multiple subsets of dictionary atoms $\mathbf{D}_H^k$, where $k = 1, \ldots, K$. Recall that, each $\mathbf{D}_H^k$ contains dictionary atoms $\mathbf{d}$ with similar HOG features. As illustrated in Fig. 1, we can reconstruct the image component I_H^k using the corresponding dictionary set $\mathbf{D}_H^k$.

For the task of image rain removal, one of the images I_H^k from the observed K groups would indicate the high spatial frequency rain streak pattern. To identify such patterns, we consider the variance of gradients for each dictionary atoms associated with each group, i.e., we calculate the variance of HOG features of $\mathbf{d}_i^k$ in $\mathbf{D}_H^k$. If the noise patterns of interest are the rain streaks, the edge directions of the rain streaks would be consistent throughout the patches in I_H and thus dominates one of the resulting cluster k. In this case, the variance of the atoms in that cluster would be the smallest among those across different clusters, and thus we can determine the cluster and its components corresponding to such noise patterns accordingly. Once the atoms/components associated with such noise are identified and removed, we can use the remaining atoms for reconstructing the high frequency part of the image. Adding the low spatial frequency parts I_L back to this recovered output, the denoised version of I is produced. Fig. 3 shows example rain removal results using different filtering techniques.

It is worth noting that, if the noise pattern or background texture of an input image is very similar to that of the rain streaks (even in a different orientation), we would also observe a low variance value for the associated HOG features. In this case, we do *not* expect to differentiate between these two similar textural patterns using our proposed method. For this challenging case,

(a) (b)

Fig. 6. An example of single image rain removal. (a) The input image with rain steaks presented, and (b) our rain removed output.

one should consider temporal information for solving this particular rain removal task. Since this paper focuses on the self-learning algorithms for single-image decomposition and its potential applications, tasks beyond single-image processing would be out of the scope of this paper.

V. Application to Single Image Denoising

A. Image Denoising

The goal of image denoising is to remove unstructured or structured noise from an image which is acquired in the presence of an additive noise [23]. Numerous approaches have been proposed to address this problem in the literature [7], [8], [13], [24]. Extended from image denoising, algorithms have also been proposed for addressing particular image processing tasks. An example is bilateral filtering [12], which performs image denoising via Gaussian blur while being able to preserve the edge information.

Recently, the use of sparse and redundant representations has been successfully applied to address this task [6]–[8]. With a predetermined dictionary or the one learned from the input image itself, one can effectively recover the denoised version. A representative sparse-representation based denoising work is the K-SVD approach [6], [7]. Once the standard deviation of the Gaussian noise is given, very promising denoising results were reported in [6], [7]. Another popular method is BM3D (block-matching and 3D filtering) [13], which is also on the image sparse representation in the transformed domain. Similar to K-SVD, BM3D also requires the prior knowledge of the standard deviation of the Gaussian noise.

B. Our Proposed Method for Removing Gaussian Noise

Besides rain removal, we further apply our proposed decomposition method for removing Gaussian noise from input images. It is worth noting that, unlike K-SVD or BM3D, we do not need the standard deviation of such noise patterns to be given in advance, which makes our method more practical for real-world applications.

Similar to rain removal, we first decompose the input I into I_L and I_H. Fig. 4 shows examples of producing I_H using different filtering techniques. Once I_H is obtained, we learn the dictionary $\mathbf{D}_H$ and extract the HOG features for each atom $\mathbf{d}_i$. We note that, while HOG is not expected to describe the Gaussian noise, the presence of such noise would result in HOG features in which each bin/attribute is not distinguishable. On the other hand, for noise-free dictionary atoms, we still observe dominant attributes in their HOG features. As a result, the use of HOG still allows us to perform clustering of dictionary atoms. In other words, even the standard deviation of the Gaussian noise is not given, we are still able to identify the image component which corresponds to the presence of such noise using our decomposition and clustering framework. Once this noise component is identified and disregarded, we can reconstruct the image using the remaining HF components and I_L, and example results are shown in Fig. 5.

TABLE II
Performance Comparisons (in Terms of PSNR) of Bilateral-Filtering Based Rain Removal Methods

	Bilateral	MCA-based	Context-based	Ours w/ Bilateral
Fig. 3	20.07	21.66	21.29	21.78
Fig. 7	19.21	19.56	19.59	19.58
Fig. 8	24.26	24.50	24.43	24.62

TABLE III
PSNR Comparisons of Rain Removal Results Using Different Denoising Techniques

	K-SVD	Ours w/ K-SVD	BM3D	Ours w/ BM3D
Fig. 3	20.56	21.90	20.96	21.59
Fig. 7	19.43	19.87	19.50	19.72
Fig. 8	24.65	24.94	24.47	24.88

VI. Experiments

To evaluate the performance of our proposed method, we conduct experiments for addressing two single-image denoising tasks: rain removal and denoising (with Gaussian noise). We consider the patch size of each image as 16×16 pixels, and the number of dictionary atoms $M = 1024$. As suggested in [17], the regularization parameter λ and the maximum sparsity value for the OMP algorithm are set as 0.15 and 10, respectively. For LPF preprocessing techniques, we have the spatial and intensity-domain standard deviations for bilateral filtering as 6 and 0.2, respectively. All images are of size 256×256 pixels in our experiments.

A. Performance Evaluation On Single Image Rain Removal

We collect several synthetic rain images from the Internet or the photo-realistically rendered rain video frames provided in [21], and thus we have ground-truth images without rain streaks presented for PSNR calculation. To evaluate the performance of our proposed method for rain removal, we compare our method with bilateral filtering (denoted by "Bilateral") [12], K-SVD [7], and BM3D [13] denoising algorithms. We set large standard deviation values $\sigma = 25$ and 35 for K-SVD and BM3D algorithms, respectively. We note that, during the preprocessing stage of our framework, larger σ values allow us to remove high spatial frequency patterns including possible rain streaks from the low spatial frequency parts of the input image. We do not (and it is not possible) fine tune such parameters for removing the rain streaks only.

In addition to the above methods, we consider two of our prior rain removal works: MCA-based rain removal (denoted

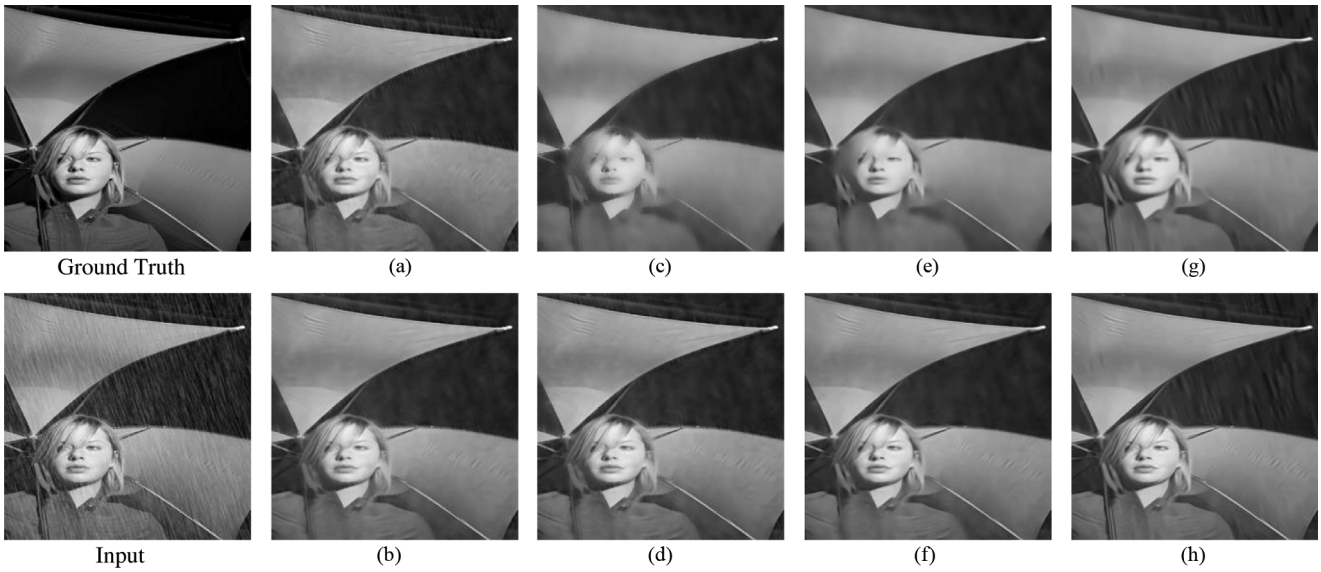

Fig. 7. Example rain removal results. Note that the input image is the noisy version of ground truth image with rain streaks presented. Rain removal outputs are produced by the methods of (a) Context-based [10], (b) MCA-based [9], (c) Bilateral [12], (d) ours with Bilateral, (e) K-SVD [7], (f) ours with K-SVD, (g) BM3D [13], and (h) ours with BM3D.

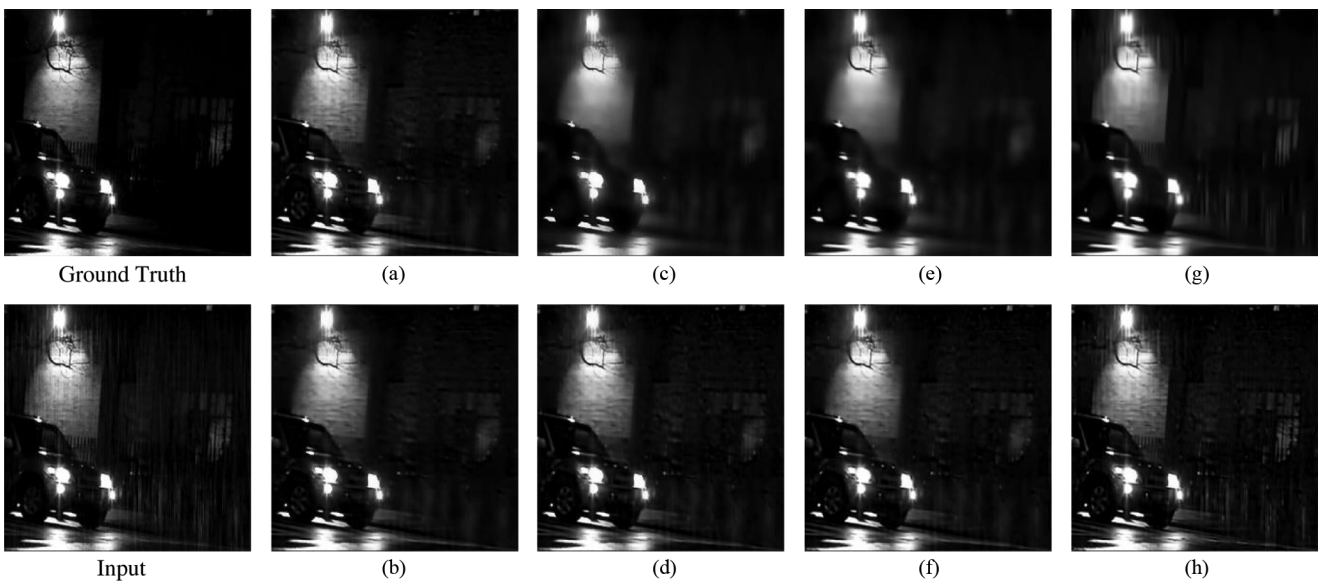

Fig. 8. Example rain removal results. Note that the input image is the noisy version of ground truth image with rain streaks presented. Rain removal outputs are produced by the methods of (a) Context-based [10], (b) MCA-based [9], (c) Bilateral [12], (d) ours with Bilateral, (e) K-SVD [7], (f) ours with K-SVD, (g) BM3D [13], and (h) ours with BM3D.

by "MCA-based") [9] and rain removal via common context pattern discovery (denoted by "Context-based") [10]. These two methods can be considered as bilateral-filtering based methods, since they require a LPF stage with a bilateral filter.

Table II lists the PSNR values of different bilateral-filtering based methods over three different rain images. From this table, we see that our proposed method achieved the highest or comparable PSNR values among different approaches. To show that we do not limit the use of bilateral filtering as the LPF algorithm, we further apply K-SVM and BM3D in our preprocessing stage, and compare the rain removal results with using these two denoising algorithms directly. As listed in Table III, it can be seen that our proposed method clearly improved the PSNR values than these two state-of-the-art denoising algorithms.

To qualitatively evaluate the performance, we show an example rain removal result in Fig. 6, in which an input color image and its rain removed version are presented. We note that when removing rain streaks from color images, we represent such images in the YUV space and perform denoising in the Y domain. To better visualize and to compare the results, Figs. 7 and 8 show example rain removal images in grayscale. From these figures, it can be observe that although Bilateral, K-SVD, and BM3D methods were able to remove most rain streaks, these denoising techniques inevitably disregarded image details (e.g., high spatial frequency parts). While applying these techniques in our LPF preprocessing stage, we were able to successfully identify/recover most non-rain image details and thus achieved improved visual quality.

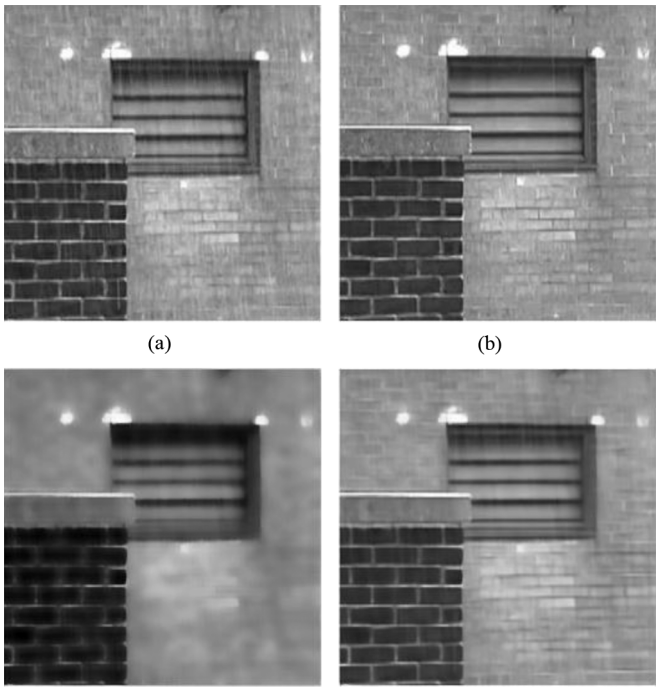

Fig. 9. Example rain removal results. (a) Original image with rain streaks presented, (b) the ground truth version of (a) (i.e., rain removed), (c) denoising output using bilateral filtering, (d) our denoising result.

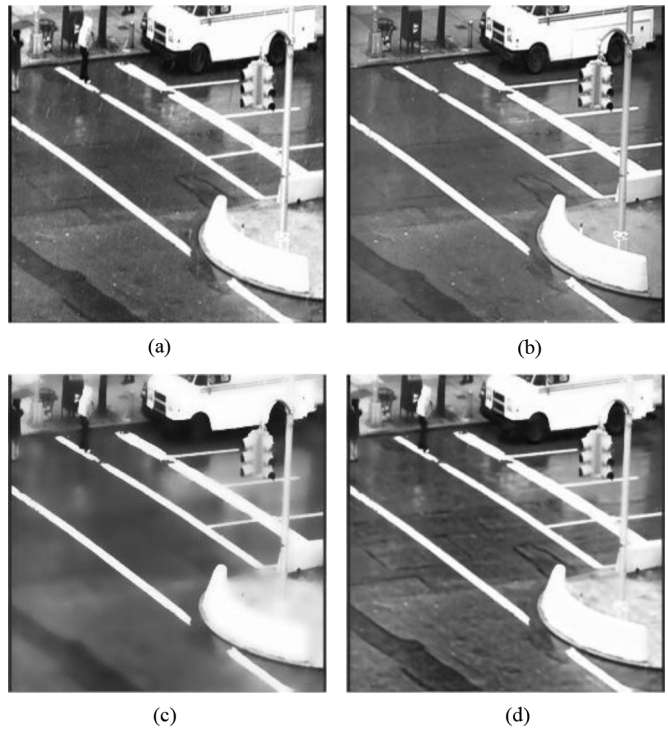

Fig. 10. Example rain removal results. (a) Original image with rain streaks presented, (b) the ground truth version of (a) (i.e., rain removed), (c) denoising output using bilateral filtering, (d) our denoising result.

It is worth noting that, although our prior MCA-based approach successfully discarded most rain streaks without significantly degrading image quality, parts of non-rain components were also removed due to the heuristic dictionary partition by *K*-means clustering algorithm. While our prior context-based method produced comparable rain removal results, it requires one to perform context-constrained image segmentation [10] on input images, and thus significantly increases the computational costs.

In addition, we perform single-image denoising experiments on real-world rainy images. In particular, we consider the image frames of the video data which were utilized in [25]. The videos in [25] were captured in real rainy scenes with static backgrounds, and the authors proposed to adjust camera parameters for removing or enhancing the presence of rain streaks. Thus, using their video data, we are able to collect real-world rainy images and the corresponding ground truth versions. We show example denoising results in Figs. 9 and 10. From these two figures, it can be seen that our approach produced satisfactory rain removal results on real-world images with rainy scenes. We note that, although bilateral filtering was able to remove high spatial frequency patterns such as rain streaks while preserving edges in Figs. 9 and 10, a large portion of image details were also removed. As a result, an automatic and self-learning based approach likes ours is preferable in removing particular noise patterns from the input image.

B. Performance Evaluation on Image Denoising

To evaluate the performance of our approach for image denoising (with Gaussian noise), we collect and conduct experiments on several images considered in [13]. We manually add Gaussian noise with $\sigma = 25$ to the input noise-free images for addressing this task. Note that if the σ for the Gaussian function is known in advance, both K-SVD and BM3D algorithms will be expected to achieve excellent denoising results. However, we assume this exact parameter choice is not known (which is practical), and we simply set large standard deviation values $\sigma = 35$ for both algorithms. Similar to the scenarios for rain removal, this would allow us to remove high spatial frequency patterns including possible Gaussian noise from the low spatial frequency parts of the input image without fine tuning the parameter σ. We also compare our algorithm with denoising methods not requiring the prior knowledge on σ for the Gaussian noise. We consider the SURE-LET algorithm [26], which relies on a purely data-adaptive unbiased estimate of the mean-squared error, so that the Gaussian noise can be removed without knowing the Gaussian parameter σ in advance.

Table IV lists the PSNR of different denoising approaches, including ours with three different LPF/denoisng techniques applied. From this table, it can be seen that our approach produced improved denoising results than the standard LPF/denoising approaches did (i.e., Bilateral filtering, K-SVD, SURE-LET, and BM3D). For qualitative comparisons, Figs. 11 and 12 show example denoising results produced by different methods. From these figures, we see that standard LPF/denoising methods were not able to achieve satisfactory results if parameters like σ are not given in advance.

Furthermore, although the SURE-LET based approach was able to outperform approaches using K-SVD for Gaussian noise removal, BM3D-based approaches still achieved the best denoising performance (i.e., BM3D with ours).

It is worth noting that, while our method quantitatively and qualitatively outperformed others, we do not need to fine-tune our approach with σ or assume such parameters are known in

TABLE IV
PERFORMANCE COMPARISONS (IN TERMS OF PSNR) OF DIFFERENT IMAGE DENOISING APPROACHES. NOTE THAT WE PRESENT OUR RESULTS USING THREE DIFFERENT LPF OR DENOISING TECHNIQUES

	Bilateral	Ours w/ Bilateral	K-SVD	Ours w/ K-SVD	SURE-LET	Ours w/ SURE-LET	BM3D	Ours w/ BM3D
Pepper	25.03	26.94	27.84	28.43	29.00	28.87	29.42	29.27
Man	22.81	24.36	24.98	25.68	27.15	26.85	26.77	27.14
Lena	24.38	25.93	27.02	27.82	28.55	28.69	29.21	29.25
House	26.06	28.51	29.79	29.79	30.29	30.34	32.29	31.87
Cameraman	24.92	26.72	28.78	29.04	28.30	28.43	28.45	28.88
Boat	24.18	25.47	25.65	26.29	27.25	27.34	27.48	27.81
Barbara	23.87	25.45	26.35	27.34	26.94	27.27	28.66	28.80
Average	24.47	26.20	27.20	27.77	28.21	28.26	28.89	29.01

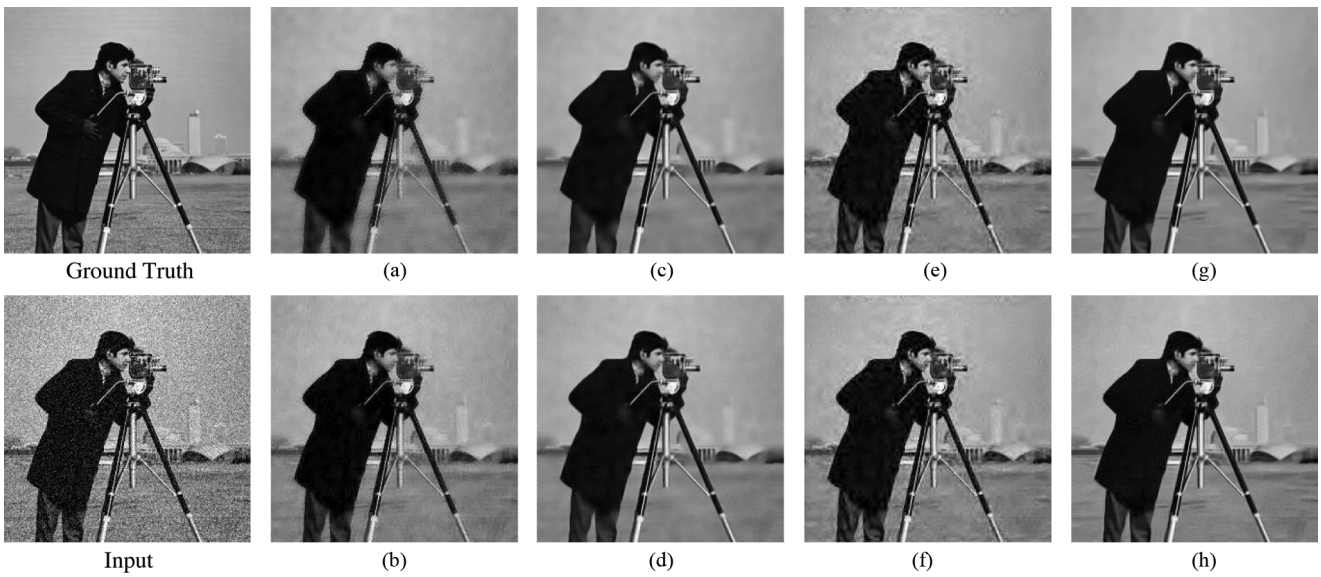

Fig. 11. Example image denoising results. Note that the input image is the noisy version of ground truth image (with Gaussian noise). Denoising outputs are produced by the methods of (a) Bilateral [12], (b) ours with Bilateral, (c) K-SVD [7], (d) ours with K-SVD, (e) SURE-LET [26], (f) ours with SURE-LET, (g) BM3D [13], and (h) ours with BM3D.

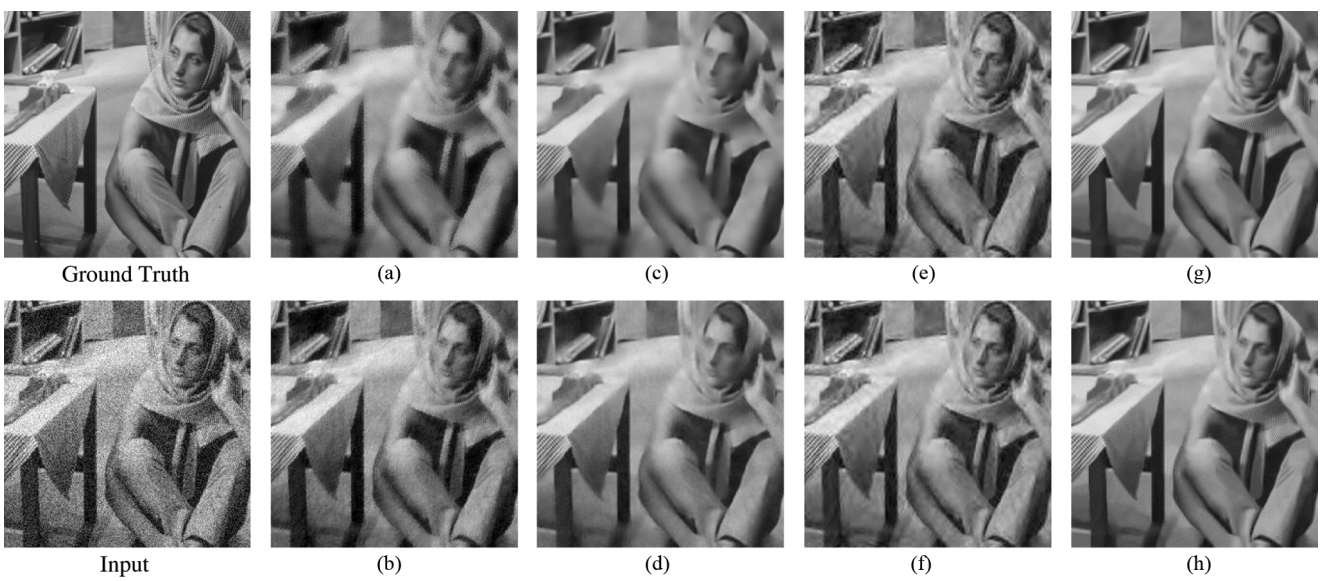

Fig. 12. Example image denoising results. Note that the input image is the noisy version of ground truth image (with Gaussian noise). Denoising outputs are produced by the methods of (a) Bilateral [12], (b) ours with Bilateral, (c) K-SVD [7], (d) ours with K-SVD, (e) SURE-LET [26], (f) ours with SURE-LET, (g) BM3D [13], and (h) ours with BM3D.

advance (which might not be practical). From the above experiments, we again confirm the effectiveness and robustness of our approach for image denoising, which can be integrated with existing LPF/denoising techniques in the LPF preprocessing stage. In other words, we do not limit the use of our proposed framework to any particular LPF or denoising algorithm.

Although real-time processing is not of concern of this paper, we provide the remarks on computation time for different

learning stages of our proposed framework as follows. In our proposed method, it takes about 100 seconds to perform denoising for an input image of 256 x 256 pixels. In particular, it takes about 3 seconds to perform bilateral filtering (i.e., identifying potential high-frequency noise patterns), 1 minute for learning the sparse-representation based dictionary, 30 seconds for performing affinity propagation to identify image components of interest, and 5 seconds for reconstructing the image output. We note that, the above runtimes were obtained on an Intel Quad Core 2 PC with 2.66 GHz processors and 4G RAM.

VII. Conclusion

In this paper, we presented a learning-based image decomposition framework for single image denoising. The proposed framework first observes the dictionary atoms from the input image for image representation. Image components associated with different context information will be automatically learned from the grouping of the derived dictionary atoms, which does not need the prior knowledge on the type of images nor the collection of training image data. To address the task of image denoising, our proposed method is able to identify image components which correspond to undesired noise patterns. Experiments on two types of single image denoising tasks (with structured and unstructured noise) confirmed the use of our proposed method, which was shown to quantitatively and qualitatively outperform existing denoising approaches.

References

[1] 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.

[2] J. Bobin, J. L. Starck, J. Fadili, and Y. Moudden, "Sparsity and morphological diversity in blind source separation," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2662–2674, Nov. 2007.

[3] F. G. Meyer, A. Z. Averbuch, and R. R. Coifman, "Multilayered image representation: application to image compression," *IEEE Trans. Image Process.*, vol. 11, no. 9, pp. 1072–1080, Sep. 2002.

[4] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 882–889, Aug. 2003.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1742–1755, Apr. 2012.

[10] D.-A. Huang, L.-W. Kang, M.-C. Yang, C.-W. Lin, and Y.-C. F. Wang, "Context-aware single image rain removal," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Melbourne, Australia, Jul. 2012, pp. 164–169.

[11] 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.

[12] 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.

[13] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3d transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[14] 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. 13, pp. 607–609, Jun. 1996.

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

[16] 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.

[17] 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, 2010.

[18] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.

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

[20] 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.

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

[22] 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.

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

[24] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Apr. 2012.

[25] 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.

[26] T. Blu and F. Luisier, "The sure-let approach to image denoising," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2778–2786, Nov. 2007.

De-An Huang received the B.S. degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan in 2012. During his B.S. study, he was supported in part by the Research Center for Information Technology Innovation (CITI), Academia Sinica, Taiwan, and worked on research topics of image super-resolution. Since 2012, he joins CITI as a full-time research assistant. His research interests are in the areas of image processing, computer vision, and machine learning.

Li-Wei Kang (S'05-M'06) received his 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 February 2013, he has been with the Graduate School of Engineering Science and Technology-Doctoral Program, and the Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Yunlin, Taiwan, as an Assistant Professor. Before that, he worked for the Institute of Information Science, Academia Sinica (IIS/AS), Taipei, Taiwan, as an Assistant Research Scholar during 2010–2013, and as a postdoctoral research fellow during 2005–2010. 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 special session co-chair of APSIPA ASC 2012, a registration co-chair of APSIPA ASC 2013, a co-organizer of special sessions of VCIP 2011–2012, and APSIPA ASC 2011–2013. He won four paper awards presented Computer Vision, Graphics, and Image Processing Conferences, Image Processing and Pattern Recognition Society, Taiwan in 2006–2007 and 2012, respectively.

Yu-Chiang Frank Wang (M'09) received the B.S degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan in 2001. From 2001 to 2002, he worked as a research assistant at the National Health Research Institutes, Taiwan. He received his M.S. and Ph.D. degrees in Electrical and Computer Engineering from Carnegie Mellon University, Pittsburgh, USA, in 2004 and 2009, respectively.

Dr. Wang joined the Research Center for Information Technology Innovation (CITI) of Academia Sinica, Taiwan in 2009, where he holds the position as a tenure-track assistant research fellow. He leads the Multimedia and Machine Learning Lab at CITI, and works in the fields of computer vision, pattern recognition, and machine learning. He has been a regular visiting scholar of the Department of Computer Science and Information Engineering at National Taiwan University Science and Technology, Taiwan from 2010 to 2012.

Chia-Wen Lin (S'94-M'00-SM'04) received his 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 Section Manager. From April 2000 to August 2000, he was a Visiting Scholar with Information Processing Laboratory, Department of Electrical Engineering, University of Washington, Seattle, USA. He has authored or coauthored over 100 technical papers. He holds more than twenty patents. His research interests include video content analysis and video networking. Dr. Lin is an Associate Editor of the IEEE Transactions on Multimedia, IEEE Transactions on Circuits and Systems for Video Technology, and the Journal of Visual Communication and Image Representation. He is also an Area Editor of EURASIP Signal Processing: Image Communication. He served as Technical Program Co-Chair of the IEEE International Conference on Multimedia and Expo (ICME) in 2010, and Special Session Co-Chair 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 SPIE VCIP 2005. He received the Young Faculty Awards presented by CCU in 2005 and the Young Investigator Awards presented by National Science Council, Taiwan, in 2006.