Journal of
Electronic Imaging

JElectronicImaging.org

# External prior learning and internal mean sparse coding for image denoising

Qiongshuai Lyu
Min Guo
Miao Ma
Richard Mankin

Qiongshuai Lyu, Min Guo, Miao Ma, Richard Mankin, "External prior learning and internal mean sparse coding for image denoising," *J. Electron. Imaging* **28**(3), 033014 (2019), doi: 10.1117/1.JEI.28.3.033014.

# External prior learning and internal mean sparse coding for image denoising

**Qiongshuai Lyu,[a,b] Min Guo,[a,*] Miao Ma,[a] and Richard Mankin[c]**
[a]Shaanxi Normal University, School of Computer Science, Ministry of Education, Key Laboratory of Modern Teaching Technology, Xi'an, China
[b]Pingdingshan University, School of Computer Science and Technology, Pingdingshan, China
[c]US Department of Agriculture, Agricultural Research Service, Center for Medical, Agricultural, and Veterinary Entomology, Gainesville, Florida, United States

**Abstract.** Image prior and sparse coding learning methods have important uses in image denoising. Many denoising methods learn priors either from the noisy image itself or an external clean image dataset. But using only these as priors does not always reconstruct the image effectively. In addition, when the image is corrupted by noise, the local sparse coding coefficient obtained from a noisy image patch is inaccurate, restricting denoising performance. We present a noise removal framework based on external prior learning and an internal mean sparse coding method, making use of the innate sparsity and nonlocal self-similarity (NSS) of natural images. Specifically, we first obtain external priors from a clean natural image dataset by Gaussian mixture model. The external priors are applied to guide the subspace clustering of internal noisy image patches, and a compact dictionary is generated for each internal noisy patch cluster. Then an internal mean sparse coding strategy based on NSS is introduced into the sparse representation model, whose regularization parameters then are deduced through a Bayesian framework. An iterative shrinkage method is employed to solve the $l_1$-optimization problem in the sparse representation model. Application of the noise removal model to 16 test images demonstrates denoising performance exceeding other competing methods. [DOI: 10.1117/1.JEI.28.3.033014]

Keywords: image denoising; prior learning; sparse coding; dictionary learning.

Paper 180959 received Nov. 4, 2018; accepted for publication May 1, 2019; published online May 23, 2019.

## 1 Introduction

Image denoising is a problem of fundamental importance for enhancement of quality in image restoration and computer vision. Due to the current trend of decreasing size of CMOS/CCD sensors, pixels capture less light and images can be more easily corrupted by noise, which makes denoising of even greater benefit than previously for many image processing applications.

In general, the purpose of image denoising is to restore a clean image $\mathbf{X}$ from its noisy observation $\mathbf{Y} = \mathbf{X} + \mathbf{v}$, where $\mathbf{v}$ is modeled as additive white Gaussian noise (AWGN). Various denoising models have been exploited over several decades, including sparse representation,[1–6] filtering,[7,8] wavelet/curvelet,[9,10] low-rank,[11–18] gradient,[19,20] neural network,[21,22] and Markov random field methods.[23,24]

The problem of image restoration has been studied with recent success by consideration of image prior information, i.e., by learning consistent structure and texture information from the image being denoised or from other natural images. Numerous denoising models based on priors have been proposed. A common approach is to learn the priors from the given noisy image itself. For the noisy image, Dabov et al.[25] proposed the famous BM3D algorithm. This algorithm combined the similar noisy patches into 3-D cubes and performed collaborative filtering to process the 3-D cubes. Portilla et al.[26] found that the use of Gaussian scale mixtures in the wavelet domain enabled improvement of image quality. Relying on the sparsity inducing priors of the noisy image itself, a joint patch sparse and group low-rank model was proposed by Wen et al.,[17] and notable restoration effects were achieved. Gu et al.[27] proposed the weighted nuclear norm minimization (WNNM) method for image denoising, utilizing the NSS prior of the noise image itself.

Another method is to obtain image prior information from a set of external clean images and then apply the prior information to denoise the corrupted image. Jiang et al.[28] proposed a sparse model to eliminate mixed noise arising from multiple sources that used a dictionary derived from a clean image dataset. Xu et al.[29] proposed a patch-group-based NSS prior denoising (PGPD) model that learns the NSS prior from the external clean image dataset to achieve AWGN denoising. Moreover, Xu et al.[2] proposed to learn the NSS model from external data and the noisy image for real-world image denoising. Similarly, Yao et al.[30] proposed to build a principal component subdictionary from the external clean image dataset to improve the denoising effect. Liu et al.[31] proposed introducing the priors of the noise-free image into a partial differential equation to realize denoising of remote-sensing images. However, priors learned from the noisy image itself may be inaccurate owing to the corruption from impulse or other errors (e.g., analog to digital conversion and communication channel errors), whereas the priors learned from the external clean dataset may have structure and texture information inconsistent with those of the noisy image. An image prior guidance strategy is needed that mitigates the existing defects induced by noise in external or internal priors.

*Address all correspondence to Min Guo, E-mail: guomin@snnu.edu.cn

1017-9909/2019/$25.00 © 2019 SPIE and IS&T

In addition to image prior information, the sparse representation model often plays a vital role in image denoising tasks. Sparsity and NSS are two important features of natural images. The use of sparsity or NSS methods can also enhance the image denoising performance. The seminal work on this approach is the nonlocal mean (NLM) method (NSS) originally proposed by Buades et al.[32] Inspired by NSS, Aharon et al.[1] proposed a $K$-means-based singular value decomposition (K-SVD) algorithm to train an over-complete dictionary based on image sparsity and redundancy, but the sparse coding distribution of nonlocal similar patches was not considered. Gao et al.[33] proposed a sparse coding algorithm based on a fixed prelearned ridge dictionary for data denoising. Zhang et al.[34] proposed a joint learning sparse coding method to preserve the global structure of the data. To achieve high-dimensional data representation, Shu et al.[35] proposed a structure-preserving sparse coding strategy. However, although these methods imposed different constraints on the sparse representation model, the accuracy of sparse coding is rarely considered. Hence, there still remains a challenging problem of how to use the redundancy of the image/data most effectively to enhance the accuracy of sparse coding.

With the above considerations, we propose an image denoising model based on external prior learning and internal mean sparse coding (EPL-IMSC). Figure 1 charts the flow of the proposed model, which has a learning phase and a denoising phase. In the learning phase, the Gaussian mixture model (GMM) is applied to learn the external priors from a clean image dataset. For noisy image patches, the learned external priors contain abundant statistical information of the structure and texture. First, in the denoising phase, the external priors are used to guide the clustering of noisy patches. The corresponding compact dictionary for each cluster is generated by principal component analysis (PCA). In this way, we avoid generating an over-complete dictionary like K-SVD[1] because many dictionary atoms are not adaptive to the structure and texture of the given noisy patch. Second, the internal mean sparse coding strategy based on NSS is introduced into the sparse representation model and image denoising is realized. Note that there are two differences from the PGPD[29] approach: (1) PGPD obtains the dictionary directly from a clean image dataset, whereas our method is to learn external priors, and then exploit the external priors to guide the clustering of internal noisy image patches. (2) In PGPD, only NSS is considered. In our model, we consider both local sparsity and NSS. Extensive experiments indicate that the peak-signal-to-noise ratio (PSNR) and structural similarity index (SSIM) of the proposed model outperform many state-of-the-art algorithms.

The work in this paper makes three main contributions to the field:

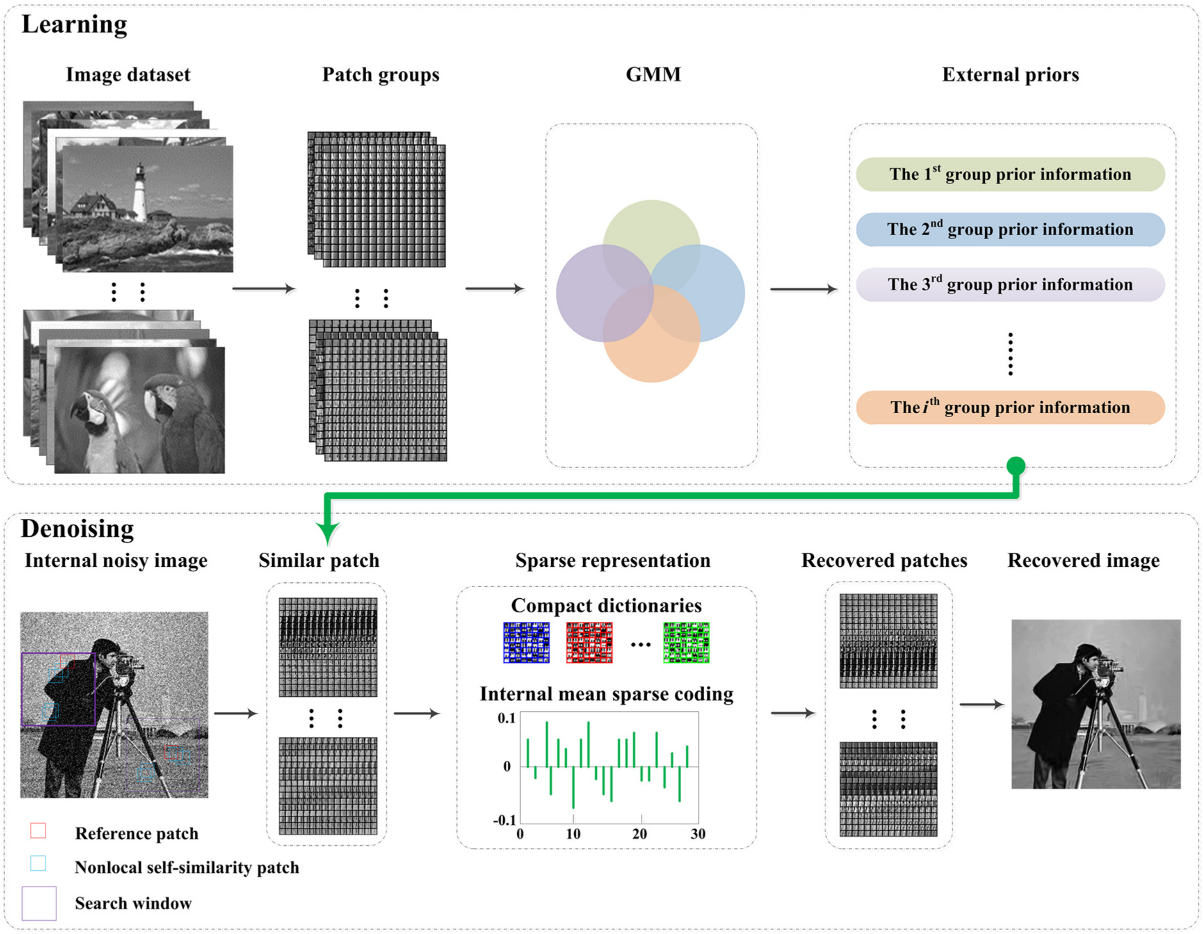

**Fig. 1** The flowchart of the EPL-IMSC model.

1. A strategy is developed for generating compact dictionaries under the guidance of external priors. The external priors are obtained from a clean image dataset by GMM. The external priors are used to guide the clustering of internal noisy patches. A corresponding compact dictionary is generated by PCA on each noisy patch cluster.
2. The internal mean sparse coding strategy based on NSS is introduced into the sparse representation model. The regularization parameters of the model are determined under the Bayesian framework. In addition, we also present a simple iterative shrinkage algorithm for solving $l_1$-optimization problem.
3. External prior learning and internal mean sparse coding are unified into a single noise removal framework.

This paper is structured as follows. Section 2 presents how to obtain external priors from a clean image dataset and generate corresponding compact dictionaries. Section 3 introduces the internal mean sparse coding method, the mechanism for determining regularization parameters under the Bayesian framework, and a simple iterative shrinkage algorithm. The experimental evaluation and performance results are described in Sec. 4, and conclusions are presented in Sec. 5.

## 2 External Prior Learning and Dictionary Generation

### 2.1 *Patch Group*

The external priors are learned from a clean image dataset. Figure 2 lists a clean set of nature images selected from Kodak PhotoCD dataset.[36] Each image is divided into $p \times p$ size patches (e.g., $p = 8$). From Fig. 2, one can see that these images contain different regions with various structures and textures. In order to make the external priors more accurate, these patches are divided into three different region sets according to the variance $\sigma^2$: areas with variance $\sigma^2 < 0.003$ are smooth regions, those with variance $0.003 < \sigma^2 < 0.03$ are structural regions, and those with variance $\sigma^2 > 0.03$ are textural regions. Then we employ the block matching algorithm to group these image patches in each region set. For each patch, the $M$ most similar patches form a patch group. Each patch group is expressed as a column vector, denoted by $\{x_m\}_{m=1}^{M}$, where $x_m \in \Re^{p^2 \times 1}$ is a patch vector.

### 2.2 *External Priors*

Suppose that $N$ patch groups are extracted from the images above, denoted as $X_n = \{x_{n,m}\}_{m=1}^{M}$, $n = 1, 2, \ldots, N$, where $x_{n,m}$ represents the $m$'th patch in the $n$'th patch group. These patch groups contain abundant structure and texture information with NSS properties. An important concern is how to obtain external priors from patch groups $X_n$. Previously, GMM was applied to model image patch priors.[37,38] Inspired by this, we design to apply GMM to learn external priors based on patch groups. We assume that each patch group $X_n$, $n = 1, 2, \ldots, N$ comes from $K$ different Gaussian subspaces. For each patch group $X_n$, the latent variable $\gamma_{n,k}$, $n = 1, \ldots, N$, $k = 1, \ldots, K$ is introduced, which means that if $x_n$ comes from the $k$'th Gaussian component $\gamma_{n,k} = 1$, otherwise $\gamma_{n,k} = 0$. Hence, the log-likelihood function with $K$ Gaussian components can be expressed as

$$\ln L = \sum_{n=1}^{N} \ln\left\{\sum_{k=1}^{K} \pi_k \prod_{m=1}^{M} \mathcal{N}(x_{n,m}|\theta_k)\right\}, \tag{1}$$

where $\sum_{k=1}^{K} \pi_k = 1$, $\mathcal{N}(x_{n,m}|\theta_k)$ is the Gaussian distribution of the $n$'th patch group, and $\theta_k = (\mu_k, \sigma_k^2)$.

An algorithm[39] comprising iterations of expectation followed by maximization stages is applied to optimize Eq. (1). In the expectation stage, we calculate $E(\gamma_{n,k}|X_n, \theta_k)$, denoted by

$$\hat{\gamma}_{n,k} = \frac{\pi_k \prod_{m=1}^{M} \mathcal{N}(x_{n,m}|\theta_k)}{\sum_{k=1}^{K} \pi_k \prod_{m=1}^{M} \mathcal{N}(x_{n,m}|\theta_k)}. \tag{2}$$

In the maximization stage, we have

$$\hat{\mu}_k = \frac{\sum_{n=1}^{N} \hat{\gamma}_{n,k} \sum_{m=1}^{M} x_{n,m}}{\sum_{n=1}^{N} \hat{\gamma}_{n,k}}, \tag{3}$$

$$\hat{\sigma}_k^2 = \frac{\sum_{n=1}^{N} \gamma_{n,k} \sum_{m=1}^{M} (x_{n,m} - \hat{\mu}_k)(x_{n,m} - \hat{\mu}_k)^{\mathrm{T}}}{\sum_{n=1}^{N} \hat{\gamma}_{n,k}}, \tag{4}$$

$$\hat{\pi}_k = \frac{\sum_{n=1}^{N} \gamma_{n,k}}{N}. \tag{5}$$

Through iteration, the parameters of the GMM are continuously updated, which ensures the convergence of the log-likelihood function. Figure 3 displays an example of the convergence curves obtained when the number of Gaussian components is $K = 72$ or $K = 200$ in the learning phase of the EPL-IMSC model. Note that instead of arduously

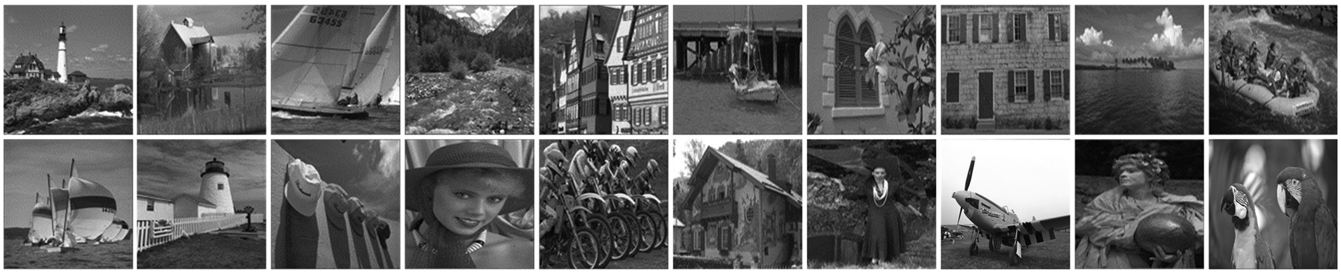

**Fig. 2** A clean image dataset.

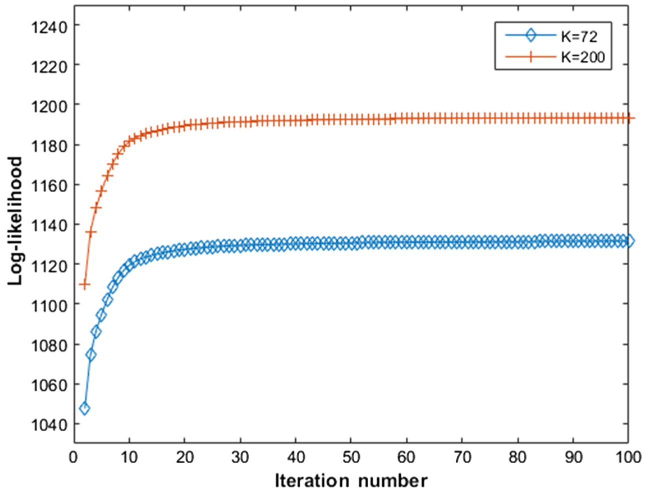

**Fig. 3** The convergence curve in the learning phase of EPL-IMSC model.

obtaining the dictionary of this patch group directly, separate compact subdictionaries are obtained from similar groups of noisy patches as described next.

### 2.3 *Dictionary Generation*

For a noisy image, patch group $\mathbf{Y} = \{y_m\}_{m=1}^{M}$ is constructed by using a block matching algorithm. As in Ref. 38, it is assumed that the image $y$ is corrupted by AWGN with variance $\sigma^2$ and mean $\mu = 0$. Then the covariance matrix of the $k$'th Gaussian component is $\sigma_k^2 + \sigma^2\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. By calculating the maximum *a posterior* (MAP), the probability that the patch belongs to the $k$'th component can be obtained. We have

$$P(k|\mathbf{Y}) = \frac{\prod_{m=1}^{M} \mathcal{N}(y_m|\theta_k)}{\sum_{i=1}^{K} \prod_{m=1}^{M} \mathcal{N}(y_m|\theta_i)}, \tag{6}$$

where $\theta_k = (\mathbf{0}, \sigma_k^2 + \delta^2\mathbf{I}), \theta_i = (\mathbf{0}, \sigma_i^2 + \delta^2\mathbf{I}), i = 1, 2, \dots, K$. Equation (6) represents the probability that $\mathbf{Y}$ belongs to the corresponding Gaussian component. Hence, the maximum value of $P(k|\mathbf{Y})$ is considered to be the cluster to which the noisy patch belongs. In other words, the noisy patches are assigned to different clusters under the guidance of external priors. The problem becomes how to generate a compact dictionary for each noisy patch cluster. Although the corresponding dictionary can also be obtained by singular value decomposition (SVD) of the covariance matrix of each Gaussian component, it cannot accurately represent the structure and texture of the noisy patches. Therefore, we propose to use a PCA strategy to generate dictionaries for noisy patch clusters. For each cluster, it is assumed that this cluster contains $Q$ similar patches, and the covariance matrix of the patch cluster can be calculated as

$$\omega = \frac{1}{Q}\sum_{q=1}^{Q} \mathbf{Y}\mathbf{Y}^{\mathrm{T}}, \tag{7}$$

where $\omega$ is a symmetric matrix. Let $\mathbf{\Omega} = \omega - \delta^2\mathbf{I}$, by SVD, the eigenvector matrix $\mathbf{D}$ and diagonal eigenvalue matrix $\mathbf{\Lambda}$ of $\mathbf{\Omega}$ are obtained

$$\mathbf{\Omega} = \mathbf{D}\Lambda\mathbf{D}^{\mathrm{T}}. \tag{8}$$

The eigenvector matrix $\mathbf{D}$ extracts the main structural and texture features from noise patch clusters, and the diagonal eigenvalue matrix represents the importance of the eigenvector. The larger the eigenvalue is, the more important the corresponding eigenvector is. Consequently, the principal statistical structure and texture features of this cluster can be represented by the $q$ top eigenvectors of $\mathbf{D}$, $q < Q$. The corresponding dictionary $\mathbf{\Phi}$ can be expressed as

$$\mathbf{\Phi} = \{d_1, d_2, \dots, d_q\}. \tag{9}$$

In this way, each noisy patch cluster has a compact dictionary, and there is no need to generate an over-complete dictionary. The corresponding compact dictionaries of four noisy patch clusters are shown in Fig. 4. For each dictionary, one can see that the dictionary atoms are different. Therefore, these compact dictionaries can better represent the characteristics of the noisy patch clusters.

## 3 Internal Mean Sparse Coding

In recent years, the sparse representation model has achieved good results in image denoising. It has been found that the human visual system can represent a natural image in a neural form of sparse coding.[40] Mathematically, the original image $\mathbf{X}$ can be expressed as $\mathbf{X} = \mathbf{D}\alpha_x$, where $\mathbf{D}$ is a dictionary and $\alpha_x$ is sparse coding, i.e., a linear combination of a small number of atoms from $\mathbf{D}$ that closely represents $\mathbf{X}$. In order to recover the latent clean image $\mathbf{X}$ from the given noise image $\mathbf{Y}$, this problem can be expressed as following a $l_0$-minimization problem:

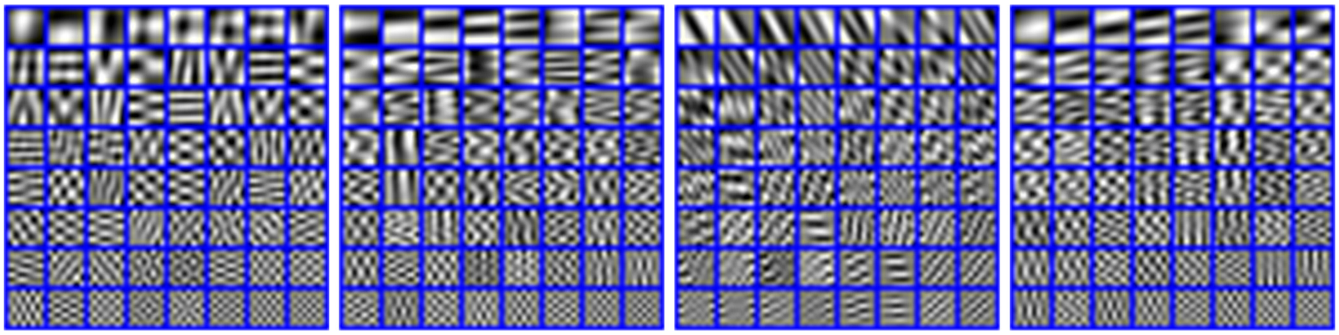

**Fig. 4** Dictionaries of four noisy patch clusters.

$$\alpha_{\mathbf{y}} = \text{argmin}_{\alpha}\|\boldsymbol{\alpha}\|_0, \quad \text{subject to } \|\mathbf{Y}-\mathbf{D}\boldsymbol{\alpha}\|_2^2 \le \epsilon, \tag{10}$$

where $\epsilon$ is a very small constant used to control the approximation error, and $\|\cdot\|_0$ counts the number of nonzeros in $\boldsymbol{\alpha}$. In practice, the $l_0$-minimization problem of Eq. (10) is nonconvex, and it is an NP-hard problem. The common practice is to replace $l_0$-norm with $l_1$-norm, which transforms the nonconvex optimization problem into convex optimization problem. The sparse coding problem with minimum $l_1$-norm is usually transformed into Lagrange form. Hence, Eq. (10) can be reformulated as

$$\alpha_{\mathbf{y}} = \arg\ \text{min}_{\alpha}\frac{1}{2}\|\mathbf{Y}-\mathbf{D}\boldsymbol{\alpha}\|_2^2 + \tau\|\boldsymbol{\alpha}\|_1, \tag{11}$$

where $\tau$ denotes the regularization parameter, which is used to achieve a balance between the data-fidelity term and the regularization term. At present, there are many effective methods to solve $l_1$-norm minimization problem, such as the iterative threshold algorithm[41,42] and the Bregman algorithm,[43] or to find an alternative strategy to avoid underestimation characteristic of $l_1$-norm regularization.[44] By solving the minimization problem of Eq. (11), we can reconstruct the latent clean image $\mathbf{X}$ with $\mathbf{D}\alpha_y$. For a given dictionary, the restored image $\mathbf{D}\alpha_y$ should be infinitely approximate to the clean image $\mathbf{D}\alpha_x$ (i.e., $\mathbf{D}\alpha_x \approx \mathbf{D}\alpha_y$). However, $\alpha_y$ comes from sparse coding of the noisy patch instead of the latent clean patch, which leads to the result that $\alpha_y$ inaccurately represents $\alpha_x$. The issue then becomes how to exploit the similarity of nonlocal patches to make $\alpha_y$ very close to $\alpha_x$.

### 3.1 *Mean Sparse Representation Model*

In Eq. (11), only local sparsity is considered without paying attention to nonlocal self-similarity. More recently, local sparsity and nonlocal self-similarity have been simultaneously applied for image processing problem.[6,15] Intuitively, using the similarity of nonlocal patches, it is possible to make $\alpha_y$ very close to $\alpha_x$ (i.e., $\alpha_x \approx \alpha_y$). The problem, however, is that $\alpha_x$ is unknown. If $\alpha_x$ can be estimated from nonlocal similar noisy patches, we can reconstruct the latent clean image. Assuming that $\alpha_x$ obeys a random distribution, the better estimate of $\alpha_x$ is its mean $E(\alpha_x)$ [i.e., $\hat{\alpha}_x = E(\alpha_x)$]. In general, $E(\alpha_x)$ can be approximate by $E(\alpha_y)$. Hence, we let $\hat{\alpha}_x \approx E(\alpha_y)$, which means that $\hat{\alpha}_{\mathrm{x}}$ is very close to the distribution mean of $\alpha_y$. The mean constraint is introduced into Eq. (11), which can be expressed as

$$\begin{aligned}\alpha_y = \arg\ \text{min}_{\alpha}&\frac{1}{2}\|\mathbf{Y}-\mathbf{D}\boldsymbol{\alpha}\|_2^2\\ &+\tau\|\boldsymbol{\alpha}\|_1 \quad \text{subject to } \|\boldsymbol{\alpha}-\boldsymbol{\beta}\|_p < e,\end{aligned} \tag{12}$$

where $\boldsymbol{\beta} = E(\alpha_y)$, $e$ is a very small constant. For each patch $x_i$, we select the $T$ top patches that are most similar to $x_i$, denoted by $x_{i,t}$, $t = 1, \dots, T$, and use the NLM method[28,32] for estimating $\boldsymbol{\beta}$, $\boldsymbol{\beta}_i = \sum_{t\in T} w_{i,t}\alpha_{i,t}$, where $w_{i,t}$ is the weight and $\alpha_{i,t}$ is sparse coding of $x_{i,t}$. The $w_{i,t}$ is inversely proportional to the Euclidean distance between $x_i$ and $x_{i,t}$, $w_{i,t} = \exp(-\|x_i - x_{i,t}\|_2^2/h)/\mathbf{W}$, where $\mathbf{W}$ is a normalization factor and $h$ is a preset scalar. The Lagrange form of Eq. (12) can be formulated as

$$\alpha_y = \text{argmin}_{\alpha}\left\{\|\mathbf{Y}-\mathbf{D}\boldsymbol{\alpha}\|_2^2 + \tau\|\boldsymbol{\alpha}\|_1 + \varepsilon\sum_i\|\boldsymbol{\alpha}_i - \boldsymbol{\beta}_i\|_p\right\}. \tag{13}$$

In Eq. (13), the first term is the data-fidelity term, the second is the local sparse coding term, and the last is the internal mean sparse coding term based on NSS. Therefore, the mean sparse representation model considers both local sparsity and NSS.

### 3.2 *Determination and Solution of Model Parameters*

There are two unknown parameters $\tau$ and $\varepsilon$ in Eq. (13). $\tau$ and $\varepsilon$ control the complexity and over-fitting of the model. In practice, it is very difficult to set these two parameters. When deriving a shrinkage function for natural image denoising, Ref. 45 discussed the method of setting wavelet coefficients using Bayesian theory. Inspired by this, we provide a Bayesian interpretation of Eq. (13) and present an explicit way to determine the two parameters. Let $\boldsymbol{\varphi} = \boldsymbol{\alpha} - \boldsymbol{\beta}$. For a given image patch $\mathbf{y}$, the MAP estimation of sparse coding $\alpha_y$ and $\varphi_y$ can be formulated as

$$(\boldsymbol{\alpha}_y, \boldsymbol{\varphi}_y) = \text{argmax}_{\alpha,\varphi}\, P(\boldsymbol{\alpha}, \boldsymbol{\varphi}|\mathbf{y}). \tag{14}$$

According to Bayesian theory

$$\begin{aligned}(\boldsymbol{\alpha}_y, \boldsymbol{\varphi}_y) &= \text{argmax}_{\alpha,\varphi}\{P(\mathbf{y}|\boldsymbol{\alpha}, \boldsymbol{\varphi})P(\boldsymbol{\alpha}, \boldsymbol{\varphi})\}\\ &= \text{argmax}_{\alpha,\varphi}\{P(\mathbf{y}|\boldsymbol{\alpha}, \boldsymbol{\varphi})P(\boldsymbol{\alpha}|\boldsymbol{\varphi})P(\boldsymbol{\varphi})\}.\end{aligned} \tag{15}$$

Assuming that the sparse coding $\boldsymbol{\alpha}$ and variable $\boldsymbol{\varphi}$ are independent of each other, we have

$$(\boldsymbol{\alpha}_y, \boldsymbol{\varphi}_y) = \text{argmax}_{\alpha,\varphi}\{P(\mathbf{y}|\boldsymbol{\alpha}, \boldsymbol{\varphi})P(\boldsymbol{\alpha})P(\boldsymbol{\varphi})\}. \tag{16}$$

By comparing with Eq. (13), the likelihood function can be expressed as

$$P(\mathbf{y}|\boldsymbol{\alpha}, \boldsymbol{\varphi}) = \frac{1}{\sqrt{2\pi}\sigma_n}\exp\left(-\frac{1}{2\sigma_n^2}\|\mathbf{y}-\mathbf{D}\boldsymbol{\alpha}\|_2^2\right). \tag{17}$$

According to Refs. 15 and 46, assuming that $P(\boldsymbol{\alpha})$ and $P(\boldsymbol{\varphi})$ follow the independent identically distributed Laplacian distribution, we have

$$P(\boldsymbol{\alpha}) = \prod_i\prod_j\frac{1}{\sqrt{2}\sigma_{i,j}}\exp\left[-\frac{|\alpha_i(j)|}{\sigma_{i,j}}\right], \tag{18}$$

$$P(\boldsymbol{\varphi}) = \prod_i\prod_j\frac{1}{\sqrt{2}\delta_{i,j}}\exp\left[-\frac{|\varphi_i(j)|}{\delta_{i,j}}\right], \tag{19}$$

where $\sigma_{i,j}$ and $\delta_{i,j}$ are the standard deviations of $\alpha_i(j)$ and $\varphi_i(j)$, $\alpha_i(j)$, and $\varphi_i(j)$ are the $j$'th elements of $\alpha_i$ and $\varphi_i$, respectively. Substituting Eqs. (17)–(19) into Eq. (13), we have

$$\alpha_y = \text{argmin}_\alpha \left\{ \|\mathbf{y} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \sum_i \sum_j \frac{2\sqrt{2}\sigma_n^2}{\sigma_{i,j}} \|\alpha_i(j)\|_1 + \sum_i \sum_j \frac{2\sqrt{2}\sigma_n^2}{\delta_{i,j}} \|\varphi_i(j)\|_1 \right\}. \tag{20}$$

By comparing Eqs. (13) and (20), we can see that $l_1$-norm should be selected to represent regularization term for mean constraint. Hence, we rewrite Eq. (13) as

$$\alpha_y = \text{argmin}_\alpha \left\{ \|\mathbf{y} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \sum_i \sum_j \tau_{i,j} \|\alpha_i(j)\|_1 + \sum_i \sum_j \varepsilon_{i,j} \|\alpha_i(j) - \beta_i(j)\|_1 \right\}. \tag{21}$$

Comparing Eq. (20) with Eq. (21), we obtain

$$\tau_{i,j} = \frac{2\sqrt{2}\sigma_n^2}{\sigma_{i,j}}, \qquad \varepsilon_{i,j} = \frac{2\sqrt{2}\sigma_n^2}{\delta_{i,j}}. \tag{22}$$

In each iteration, $\sigma_{i,j}$ and $\delta_{i,j}$ can be estimated from nonlocal self-similar image patches. The $\tau_{i,j}$ and $\varepsilon_{i,j}$ are updated through each iteration of $\alpha_y$ and $\varphi_y$. By determining the parameters $\tau$ and $\varepsilon$ in the Bayesian framework, it is possible to avoid manually setting parameters.

**Table 1** Parameter setting of the EPL-IMSC model.

| Learning phase | | Denoising phase | | |
|---|---|---|---|---|
| Patch size ($p \times p$) | Noise level $\sigma$ | $c$ | $\delta$ | $\eta$ |
| $6 \times 6$ | (0–25] | 0.19 | 0.01 | 0.30 |
| $7 \times 7$ | (25–50] | 0.13 | 0.07 | 0.38 |
| $8 \times 8$ | (50–75] | 0.12 | 0.06 | 0.35 |
| $9 \times 9$ | (75–100] | 0.05 | 0.05 | 0.50 |

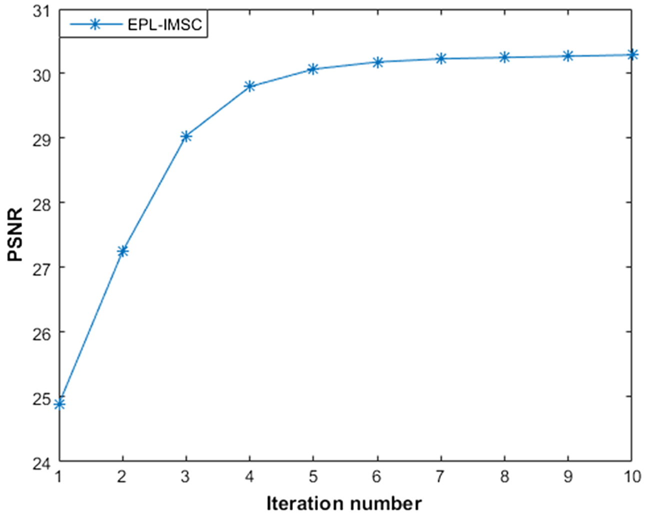

**Fig. 5** The convergence curve in the denoising phase of EPL-IMSC model.

We adopt an iterative shrinkage algorithm to solve Eq. (21). Note that subscript is omitted for brevity. Eq. (21) can be rewritten as

**Algorithm 1** EPL-IMSC denoising

```
I. Learning phase:
1. Input:
    A clean image dataset, the number of Gaussian components $K$.
2. Learning:
    While ~converged && $t <$ Iteration
        (a) E-step: calculate the response degree of the $k$'th submodel to the input data via Eq. (2);
        (b) M-step: calculation of model parameters for the next iterations via Eqs. (3)–(5);
    End
3. Output: External priors.
II. Denoising phase:
1. Input: Noisy image $y$, external priors.
2. Initialization: $\hat{x}^{(0)} = y$, $y^{(0)} = y$.
3. While $t <$ Iteration
        (a) Iterative Regularization: $y^{(t)} = \hat{x}^{(t-1)} + \delta[y - y^{(t-1)}]$;
        (b) Estimate the standard deviation $\sigma$ of noise:
        $\sigma^{(t)} = \eta \times \sqrt{\delta^2 - \|y - y^{(t-1)}\|_2^2}$;
        (c) For $i = 1, 2, \ldots, K$
            (1) Clustering the noisy patch according to the external priors via Eq. (6);
            (2) Generate compact dictionaries via Eqs. (7)and (8);
        End for
        (d) Estimate internal mean sparse coding by nonlocal similar patches;
        (e) Update the regularization parameters ($\tau$ and $\varepsilon$) via Eq. (22);
        (f) Compute $\alpha^{(k+1)}$ using the soft-thresholding operator via Eq. (27);
        (g) Restore each patch in this cluster.
    End
4. Output: The denoised image.
```

$$f(\alpha,\beta)=\|\mathbf{y}-\mathbf{D}\alpha\|_2^2+\tau\|\alpha\|_1+\varepsilon\|\alpha-\beta\|_1. \tag{23}$$

According to Ref. 42, the following surrogate function is introduced

$$\zeta(\alpha,\alpha_0)=\frac{c}{2}\|\alpha-\alpha_0\|_2^2-\frac{1}{2}\|\mathbf{D}\alpha-\mathbf{D}\alpha_0\|_2^2, \tag{24}$$

where $\alpha_0=\boldsymbol{D}^{\mathrm{T}}y$ and $c$ is a preset constant to make surrogate function $\zeta(.)$ convex. The surrogate function[47] strategy can be regarded as a proximal point method for convex optimization problems. Then we have

$$\begin{aligned}\overline{f}(\alpha,\beta,\alpha_0)&=\|\mathbf{y}-\mathbf{D}\alpha\|_2^2+\tau\|\alpha\|_1+\varepsilon\|\alpha-\beta\|_1\\&\quad+\frac{c}{2}\|\alpha-\alpha_0\|_2^2-\frac{1}{2}\|\mathbf{D}\alpha-\mathbf{D}\alpha_0\|_2^2.\end{aligned} \tag{25}$$

After some manipulation, Eq. (25) can be simplified into

$$\overline{f}(\alpha,\beta,\alpha_0)=\tau\|\alpha\|_1+\varepsilon\|\alpha-\beta\|_1+\frac{c}{2}\|\alpha-z_0\|_2^2+\text{constant}, \tag{26}$$

where $z_0=\frac{1}{c}\times\mathbf{D}^{\mathrm{T}}(\mathbf{y}-\mathbf{D}\alpha_0)+\alpha_0$. Hence, the objective function Eq. (23) can be solved by an iterative shrinkage operation. In the $(k+1)$'th iteration, we have

$$\alpha_i^{(k+1)}=\begin{cases}\mathbf{S}_{\rho_1,\rho_2,\beta_i}[z_i^{(k)}] & \beta_i\geq 0\\ -\mathbf{S}_{\rho_1,\rho_2,-\beta_i}[-z_i^{(k)}] & \beta_i<0\end{cases}, \tag{27}$$

where $\mathbf{S}_{p_1,\rho_2,\beta_i}(.)$ is the soft-thresholding operator, $\rho_1=\frac{\tau}{c}$, $\rho_2=\frac{\varepsilon}{c}$, and $z^k=\frac{1}{c}\times\mathbf{D}^{\mathrm{T}}[\mathbf{y}-\mathbf{D}\alpha^{(k)}]+\alpha^{(k)}$.

### 3.3 *Summary of the Algorithm*

The EPL-IMSC model consists of two phases. The first is the learning phase. The main purpose is to learn external priors from a clean image dataset. The second is the denoising phase. Clustering noisy patches under the guidance of external priors, we then utilize PCA to generate compact dictionaries. $\delta$ is a predetermined constant (Table 1) controlling the amount of noise feedback to the iteration. $\sigma$ is the standard deviation, which is estimated by $\sigma^{(t)}=\eta\times\sqrt{\sigma^2-\|y-y^{(t-1)}\|_2^2}$, where $\eta$ is also a predetermined constant (Table 1). The magnitude of $\sigma$ decreases gradually as the iteration continues, and the noisy image becomes clearer. Figure 5 shows the convergence curve by applying the EPL-IMCS algorithm to a noisy Lena image ($\sigma=25$) in the denoising phase.

Finally, image denoising is achieved by employing an internal mean sparse representation model.

The proposed noise removal model is summarized in Algorithm 1.

## 4 Experiment Results

All experiments are performed under the MATLAB2016b environment on a machine with Intel® Core™ i7-6700 CPU of 3.4 GHz and 32 GB RAM. To evaluate the denoising effect of the EPL-IMSC model, we performed experiments on 16 images (as shown in Fig. 6). From Fig. 6, the eight images in the first line are commonly used, the other eight images in the second line are randomly selected from the

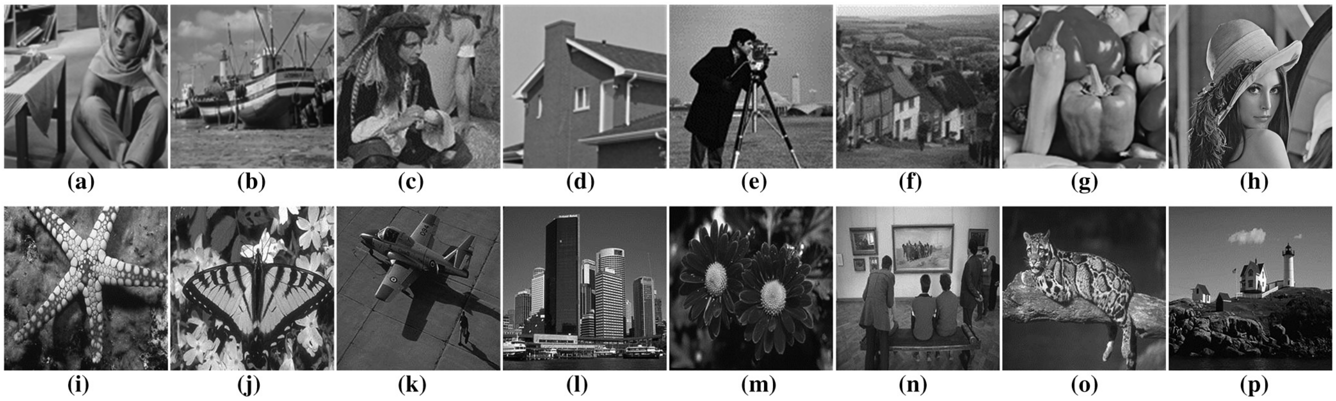

**Fig. 6** 16 test images: (a) Barbara, (b) Boat, (c) Man, (d) House, (e) C. man, (f) Hill, (g) Peppers, (h) Lena, (i) 12003, (j) 35010, (k) 37073, (l) 69007, (m) 124084, (n) 128035, (o) 160068, and (p) 228076.

**Table 2** The learning time (h) and the average PSNR/SSIM results on widely used test images.

| | | $\sigma=25$ | $\sigma=50$ | $\sigma=75$ | $\sigma=100$ |
|---|---|---|---|---|---|
| *K* | Learning time | PSNR/SSIM | PSNR/SSIM | PSNR/SSIM | PSNR/SSIM |
| 72 | 3.86 | 29.76/0.8737 | 27.70/0.7634 | 25.21/0.7089 | 23.26/0.6572 |
| 200 | 8.12 | 29.75/0.8734 | 27.71/0.7635 | 25.23/0.7090 | 23.29/0.6574 |

**Table 3** PSNR/SSIM results comparison on eight commonly used images.

| | $\sigma = 25$ | | | | | | | $\sigma = 50$ | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Images | BM3D | EPLL | FRIST | WNNM | STROLLR | PGPD | EPL-IMSC | BM3D | EPLL | FRIST | WNNM | STROLLR | PGPD | EPL-IMSC |
| Barbara | 29.79 | 29.35 | 27.64 | **30.25** | 27.97 | 29.86 | 29.40 | 27.01 | 24.85 | 27.32 | 25.77 | 24.94 | 26.81 | **27.35** |
| | 0.8874 | 0.8831 | 0.8143 | **0.8976** | 0.8256 | 0.8878 | 0.8836 | 0.7946 | 0.7031 | 0.8039 | 0.7145 | 0.7078 | 0.7803 | **0.8137** |
| Boat | 25.56 | 25.37 | 25.69 | 25.67 | 25.35 | 25.85 | **25.95** | 26.32 | 26.71 | 26.73 | **26.98** | 26.52 | 26.84 | 26.96 |
| | 0.8014 | 0.7939 | 0.8034 | 0.8047 | 0.7923 | 0.8066 | **0.8102** | 0.7053 | 0.6925 | 0.7082 | **0.7118** | 0.7084 | 0.7011 | 0.7109 |
| C. man | 29.68 | 29.43 | 29.70 | 29.84 | 29.85 | 29.27 | **30.12** | 26.14 | 26.42 | 26.10 | 26.17 | 26.33 | 26.46 | **26.51** |
| | 0.8544 | 0.8584 | 0.8591 | 0.8597 | 0.8598 | 0.8517 | **0.8602** | 0.7762 | 0.7824 | 0.7731 | 0.7764 | 0.7803 | 0.7773 | **0.7835** |
| Hill | 28.99 | 28.89 | 29.14 | 29.27 | 29.12 | **29.83** | 29.09 | 26.92 | **27.33** | 26.69 | 26.99 | 26.96 | 27.22 | 27.30 |
| | 0.7748 | 0.7679 | 0.7792 | 0.7796 | 0.7698 | **0.7821** | 0.7695 | 0.6747 | 0.6858 | 0.6639 | 0.6744 | 0.6751 | 0.6802 | **0.6861** |
| House | 32.15 | 32.48 | 32.12 | 32.49 | 32.52 | 32.91 | **32.92** | 29.49 | 29.28 | 29.12 | 29.45 | 29.53 | **29.92** | 29.65 |
| | 0.8489 | 0.8576 | 0.8472 | 0.8531 | 0.8581 | **0.8585** | 0.8582 | 0.8122 | **0.8193** | 0.8104 | 0.8077 | 0.8128 | 0.8125 | 0.8129 |
| Lena | 30.05 | 29.98 | 30.18 | 30.15 | 30.18 | 30.03 | **30.29** | 29.06 | 29.18 | 29.24 | 29.22 | 29.03 | 29.10 | **29.27** |
| | 0.8607 | 0.8599 | 0.8640 | 0.8644 | 0.8621 | 0.8592 | **0.8656** | 0.7794 | 0.7716 | 0.7832 | 0.7817 | 0.7785 | 0.7971 | **0.7983** |
| Man | 28.84 | 28.87 | 28.73 | 28.99 | 28.83 | 29.05 | **29.05** | 25.16 | 25.12 | 25.22 | 25.19 | 25.20 | **25.85** | 25.27 |
| | 0.8047 | 0.8049 | 0.8012 | 0.8110 | 0.8010 | **0.8022** | 0.8019 | 0.7056 | 0.6965 | 0.7084 | 0.6979 | 0.7031 | **0.7096** | 0.7072 |
| Peppers | 31.27 | 31.12 | 31.15 | 31.25 | 31.13 | 30.14 | **31.38** | 29.19 | 29.28 | 29.34 | **29.36** | 29.35 | 29.22 | 29.33 |
| | 0.8676 | 0.8690 | 0.8702 | 0.8724 | 0.8693 | 0.8568 | **0.8737** | 0.7936 | 0.7923 | 0.7930 | **0.7954** | 0.7948 | 0.7934 | 0.7939 |
| Average | 29.54 | 29.44 | 29.29 | 29.73 | 29.37 | 29.62 | **29.76** | 27.41 | 27.27 | 27.47 | 27.39 | 27.23 | 27.68 | **27.70** |
| | 0.8676 | 0.8690 | 0.8702 | 0.8724 | 0.8639 | 0.8568 | **0.8737** | 0.7552 | 0.7429 | 0.7555 | 0.7450 | 0.7451 | 0.7564 | **0.7634** |
| | $\sigma = 75$ | | | | | | | $\sigma = 100$ | | | | | | |
| Images | BM3D | EPLL | FRIST | WNNM | STROLLR | PGPD | EPL-IMSC | BM3D | EPLL | FRIST | WNNM | STROLLR | PGPD | EPL-IMSC |
| Barbara | 23.37 | 22.98 | 23.57 | 23.62 | 23.46 | 23.60 | **23.69** | 21.30 | 20.79 | 20.89 | 21.57 | 21.09 | 21.48 | **21.67** |
| | 0.7112 | 0.6887 | 0.7302 | 0.7421 | 0.7287 | 0.7397 | **0.7476** | 0.6430 | 0.5871 | 0.5889 | 0.6541 | 0.6263 | 0.6389 | **0.6557** |
| Boat | 25.14 | 25.05 | **25.26** | 24.87 | 25.03 | 25.10 | 25.12 | 21.65 | 21.48 | 21.83 | 22.01 | 21.98 | **22.06** | 22.03 |
| | 0.6410 | 0.6350 | **0.6457** | 0.6368 | 0.6413 | 0.6407 | 0.6410 | 0.5836 | 0.5744 | 0.5879 | 0.5902 | 0.5888 | **0.5909** | 0.5904 |
| C. man | 24.15 | 24.19 | 24.31 | 24.22 | 24.23 | **24.64** | 24.32 | 21.96 | 22.06 | 22.13 | 22.15 | 22.10 | **22.22** | 22.12 |
| | 0.7341 | 0.7313 | 0.7364 | 0.7353 | 0.7349 | **0.7376** | 0.7369 | 0.6924 | 0.6918 | 0.6925 | **0.6980** | 0.6921 | 0.6974 | 0.6928 |
| Hill | 23.76 | 23.61 | 23.49 | 23.78 | 23.68 | 23.73 | **23.82** | 23.15 | 23.28 | 23.25 | 23.23 | 23.46 | 23.56 | **23.58** |
| | 0.6118 | 0.5974 | 0.5831 | 0.6186 | 0.5987 | 0.6091 | **0.6194** | 0.5650 | 0.5679 | 0.5681 | 0.5726 | 0.5692 | 0.5615 | **0.5698** |
| House | 27.59 | 26.85 | **28.84** | 27.88 | 28.38 | 27.81 | 28.28 | 26.04 | 26.24 | **26.62** | 25.79 | 26.50 | 26.16 | 26.56 |
| | 0.7645 | 0.7533 | **0.7849** | 0.7723 | 0.7821 | 0.7709 | 0.7817 | 0.7203 | 0.7233 | 0.7301 | 0.7173 | 0.7285 | 0.7195 | **0.7305** |
| Lena | 26.26 | 25.93 | 25.95 | 26.40 | 25.58 | **26.47** | 26.44 | 24.28 | 24.23 | 24.39 | 24.44 | 24.42 | 24.19 | **24.51** |
| | 0.7516 | 0.7409 | 0.7421 | 0.7548 | 0.7373 | 0.7537 | **0.7552** | 0.7090 | 0.7085 | 0.7133 | 0.7279 | 0.7258 | 0.7065 | **0.7287** |
| Man | 24.71 | 24.82 | 24.57 | 24.88 | 24.90 | 24.36 | **24.95** | 21.83 | 21.59 | 21.50 | 21.82 | 21.93 | **22.98** | 21.92 |
| | 0.6445 | 0.6362 | 0.6478 | 0.6492 | 0.6483 | 0.6442 | **0.6501** | 0.5978 | 0.5752 | 0.5845 | 0.6045 | 0.5972 | **0.6064** | 0.5963 |
| Peppers | 24.75 | 24.62 | 24.36 | 24.95 | 25.07 | 24.85 | **25.09** | 23.28 | 23.05 | 22.84 | 23.36 | 23.55 | 23.25 | **23.57** |
| | 0.7368 | 0.7239 | 0.7209 | 0.7391 | **0.7397** | 0.7390 | 0.7392 | 0.6881 | 0.6819 | 0.6783 | 0.6904 | 0.6928 | 0.6858 | **0.6932** |
| Average | 24.97 | 24.62 | 25.04 | 25.08 | 25.04 | 25.07 | **25.21** | 22.93 | 22.84 | 22.93 | 23.05 | 23.13 | 23.24 | **23.26** |
| | 0.6994 | 0.6883 | 0.6989 | 0.7060 | 0.7014 | 0.7044 | **0.7089** | 0.6499 | 0.6388 | 0.6430 | 0.6569 | 0.6526 | 0.6509 | **0.6572** |

Berkeley segmentation dataset (BSD).[48] AWGN is added to those original images, the mean is zero and the standard deviation is 25, 50, 75, and 100, respectively.

In EPL-IMSC, there are five parameters to be set. Similar to Ref. 27, we set a larger patch size for higher noise level. For different noise levels and patch size, we empirically set parameters $c$, $\delta$, and $\eta$ as different values, as shown in Table 1. The number of the similar patches in a patch group is set to $M = 16$. The size of the search window is set to $W = 31$. We compared the proposed EPL-IMSC model with other state-of-the-art noise removal models, including BM3D,[25] EPLL,[38] FRIST,[6] WNNM,[27] STROLLR,[17] and PGPD.[29] These models are available from the author's personal home page, and all parameters are defaults.

### 4.1 *Experiments on Widely Used Test Images*

We tested the effect of different values of the Gaussian component $K$ on the denoising performance. The learning time (h) and the average PSNR/SSIM on widely used test images are listed in Table 2. From Table 2, one can see that when the number of Gaussian components $K = 200$, the performance of denoising only improve a little in the denoising phase, but it takes more learning time in the learning phase. Therefore, for all experiments, we set $K = 72$.

The PSNR/SSIM results are reported in Table 3, and we can see that the average PSNR/SSIM of the proposed model is much better than other models. Although it is slightly worse than other algorithms on individual images, it does not affect the overall effect of the EPL-IMSC model.

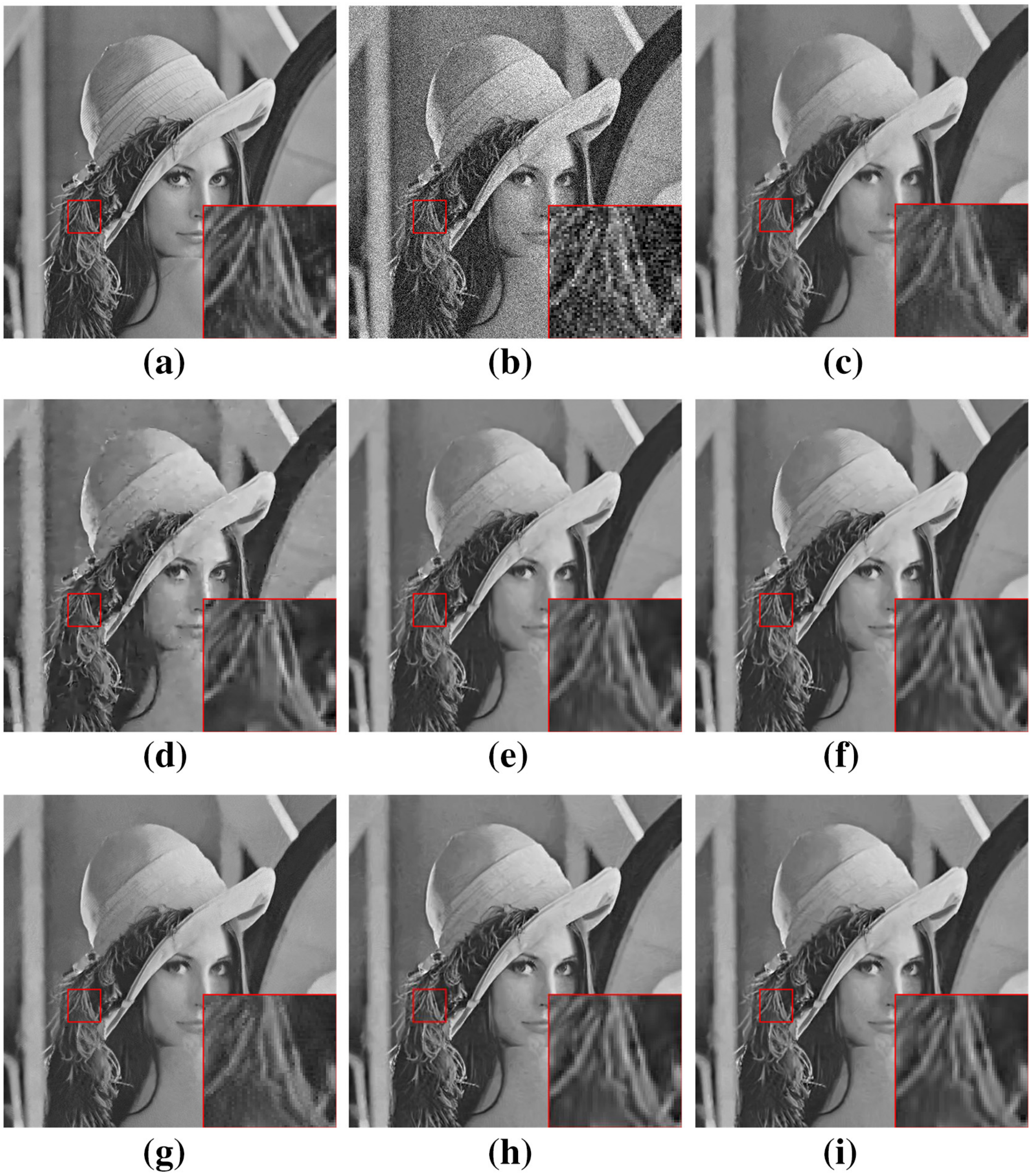

**Fig. 7** Denoising results on image "Lena" by different methods ($\sigma = 25$): (a) ground truth (PSNR/SSIM), (b) noisy image (20.18/0.2709), (c) BM3D (30.05/0.8607), (d) EPPL (29.98/0.8599), (e) FRIST (30.18/0.8640), (f) WNNM (30.15/0.8644), (g) STROLLR (30.18/0.8621), (h) PGPD (30.03/0.8592), and (i) EPL-IMSC (30.29/0.8656).

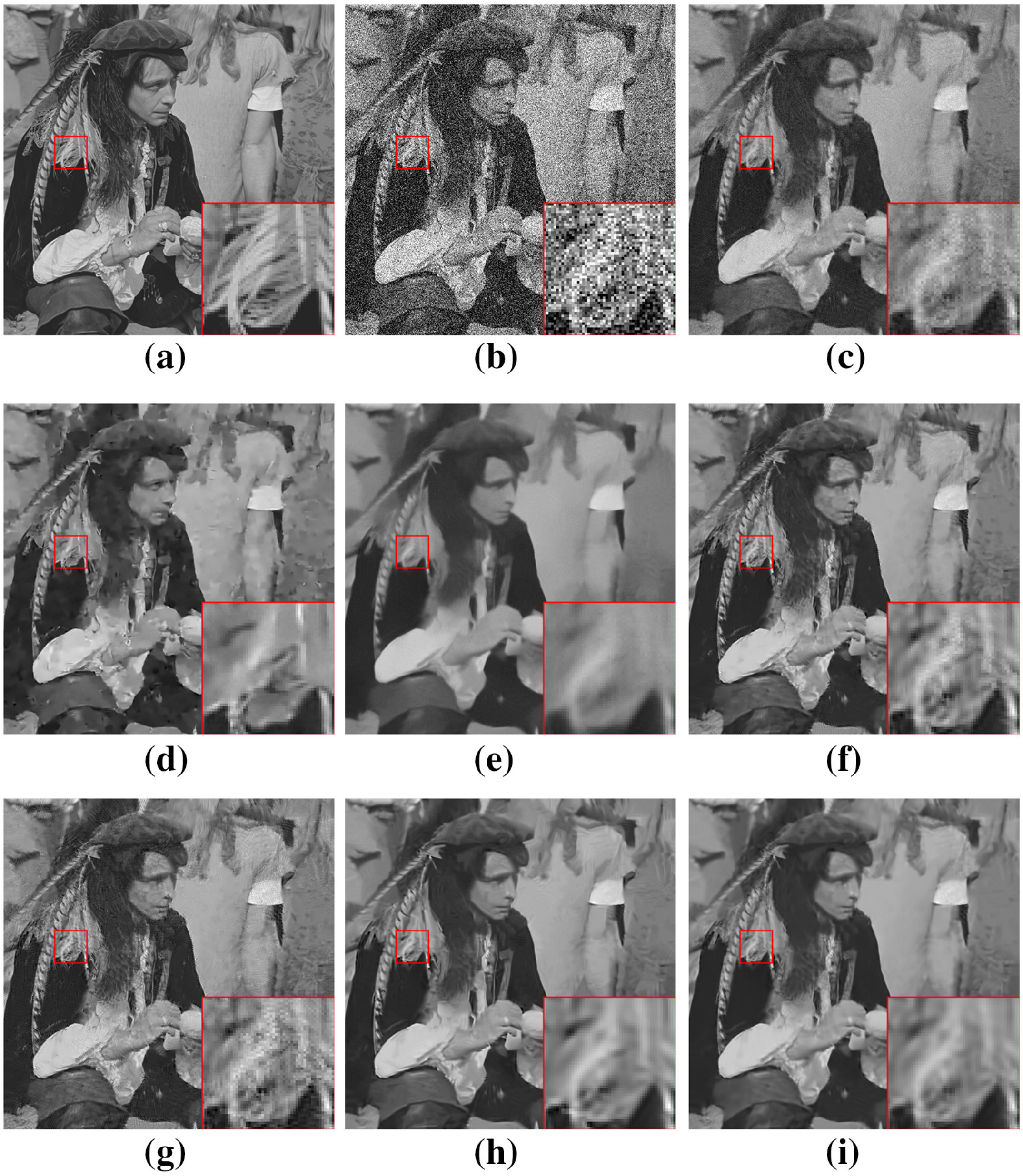

**Fig. 8** Denoising results on image “Man” by different methods ($\sigma = 50$): (a) ground truth (PSNR/SSIM), (b) noisy image (14.16/0.1407) (c) BM3D (25.16/0.7056), (d) EPPL (25.12/0.6965), (e) FRIST (25.22/0.7084), (f) WNNM (25.19/0.6979), (g) STROLLR (25.20/0.7031), (h) PGPD (25.85/0.7096), and (i) EPL-IMSC (25.27/0.7072).

The denoising effects of different noise levels are shown in Figs. 7–10. From the overall visual effect, one can see that, as the noise level increases, BM3D and EPLL blur the image texture, WNNM generates severe artifacts, and PGPD destroys the local structure, whereas FRIST and STROLLR smooth the image details and edges. By comparison, our proposed model not only effectively eliminates noise but also preserves the local details of the image.

### 4.2 *Experiments on Berkeley Segmentation Data Set*

The average PSNR/SSIM result of each method is shown in Table 4. Obviously, the average PSNR/SSIM of the proposed EPL-IMSC is clearly much better than other methods on the whole. Figures 11 and 12 show the visual effects of different denoising methods. As can be seen from these figures, all algorithms generate artifacts when the noise level is high, but the visual effect of our proposed model is more pleasant.

In addition, we also tested the EPL-IMSC model on color images and compared it with CBM3D.[49] For simplicity, we first transformed the color image from RGB to YCbCr and then tested it on the luminance component. As can be seen from first line in Fig. 13, the EPL-IMSC model restores the texture of the butterfly wings better. In the second line of Fig. 13, it is clear that the EPL-IMSC model more faithfully recovers the details of the flower.

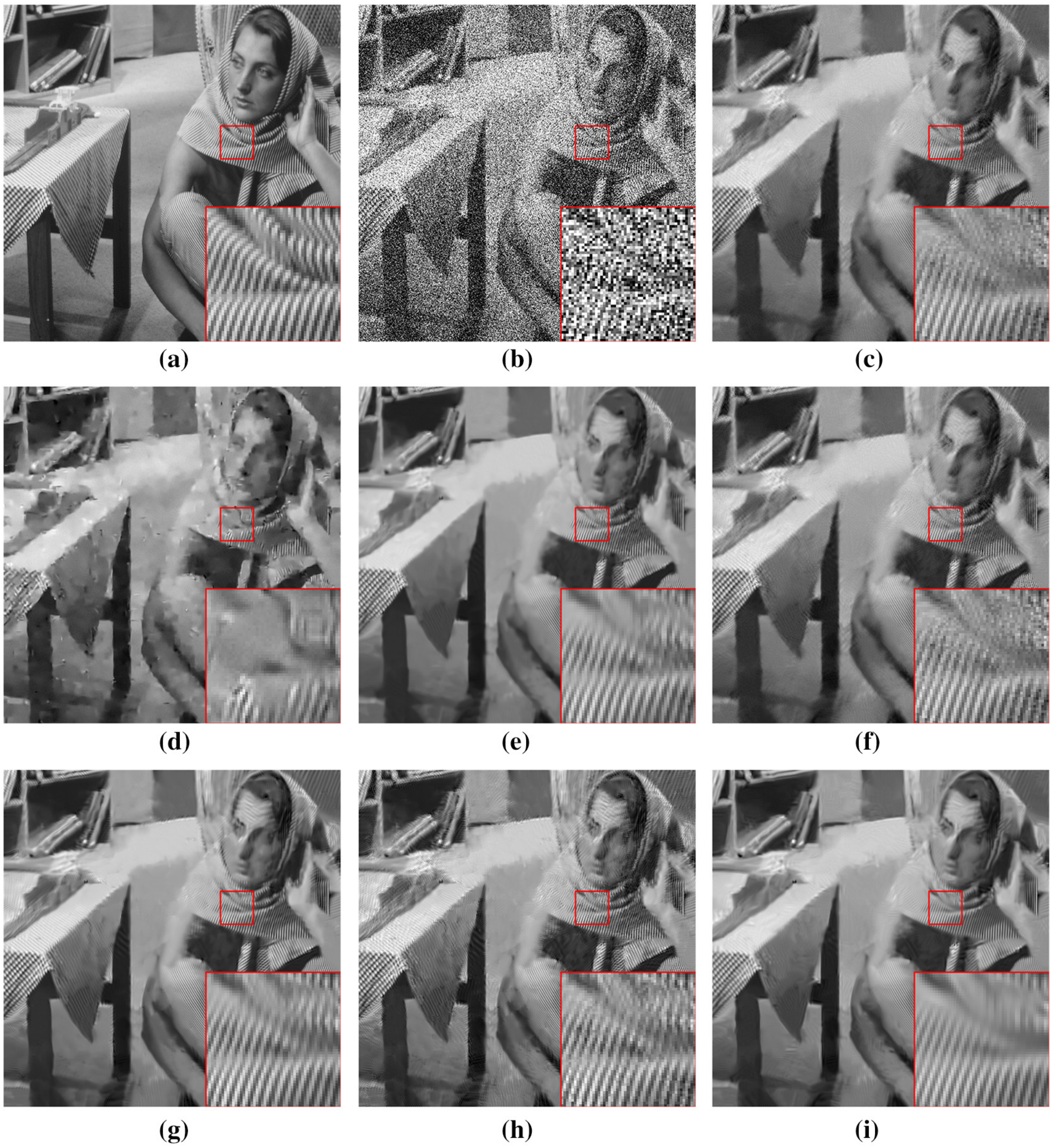

**Fig. 9** Denoising results on image "Barbara" by different methods ($\sigma = 7$): (a) ground truth (PSNR/SSIM), (b) noisy image (10.63/0.1118) (c) BM3D (25.37/0.7112), (d) EPPL (29.98/0.6887), (e) FRIST (23.57/0.7302), (f) WNNM (23.62/0.7421), (g) STROLLR (23.46/0.7287), (h) PGPD (23.60/0.7397), and (i) EPL-IMSC (23.69/0.7476).

### 4.3 *Run Time*

In addition to PSNR/SSIM, another important evaluation method for denoising is the run time. Table 5 lists the run time (s) of competing methods on 16 images. From Table 5, we can see that the run time of BM3D is the shortest, because BM3D is implemented by C++ mex-function and with parallelization. PGPD runs faster than other methods, but it is slower than BM3D. The main reason is that offline dictionaries are used in the model. In EPLL, the priors are learned beforehand, and half-quadratic splitting method can optimize cost function effectively. WNNM requires a large number of online SVD operations. The proposed EPL-IMSC is slower because it needs to train an online dictionary for each noisy patch cluster. Both STROLLR and FRIST take a long time, STROLLR involves a block coordinate descent algorithm that combines image patch sparsity and group low-rank, whereas FRIST needs to learn online synthesis dictionary, and this method involves expensive learning steps.

## 5 Conclusion

We proposed a denoising model based on external prior learning and internal mean sparse coding. The whole model can be divided into two phases: in the learning phase, we first obtained external priors from a clear image dataset. To make better use of the NSS of image patches, we exploited the GMM model to learn external priors from patch groups rather than from a single patch. In the denoising phase, external priors were utilized to guide the clustering of internal noise patches and a corresponding compact dictionary was generated for each cluster. By unbiased estimation of sparse coding, the mean sparse coding strategy was introduced into the sparse representation model. We presented a method to determine model

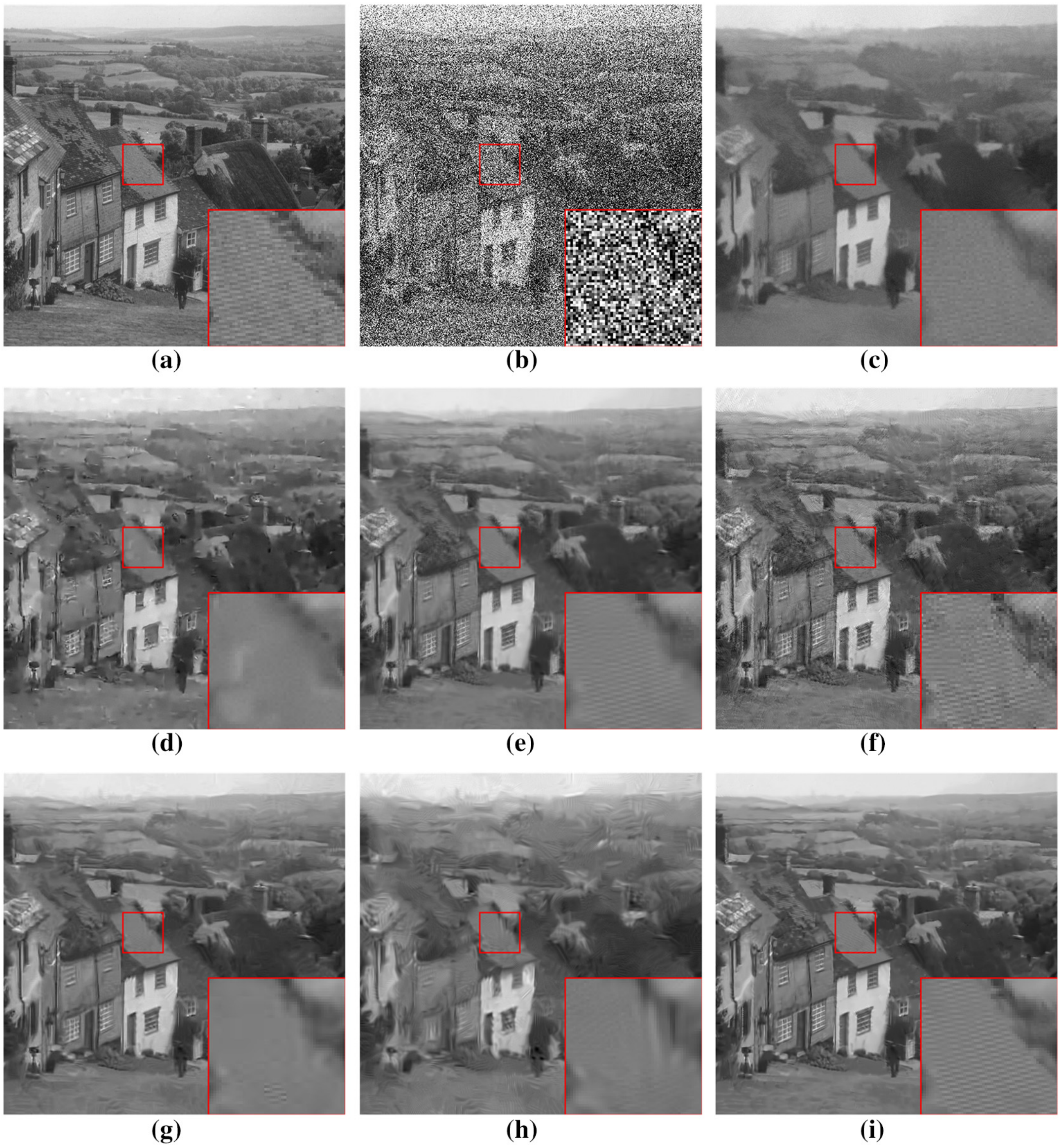

**Fig. 10** Denoising results on image "Hill" by different methods ($\sigma = 100$): (a) ground truth (PSNR/SSIM), (b) noisy image (8.14/0.0395) (c) BM3D (23.15/0.5650), (d) EPPL (23.28/0.5679), (e) FRIST (23.25/0.5681), (f) WNNM (23.23/0.5726), (g) STROLLR (23.46/0.5692), (h) PGPD (23.56/0.5615), and (i) EPL-IMSC (23.58/0.5698).

**Table 4** Average PSNR/SSIM results on randomly selected eight images from BSD.

| $\sigma$ | BM3D | EPLL | FRIST | WNNM | STROLLR | PGPD | EPL-IMSC |
|---|---|---|---|---|---|---|---|
| 25 | 27.56 | 27.48 | 27.76 | 27.81 | 27.85 | 27.74 | **27.97** |
| | 0.8235 | 0.8178 | 0.8242 | 0.8256 | 0.8247 | 0.8238 | **0.8263** |
| 50 | 24.78 | 24.81 | 24.75 | 24.76 | 24.83 | 24.86 | **24.95** |
| | 0.7528 | 0.7533 | 0.7529 | 0.7544 | 0.7561 | **0.7568** | 0.7564 |
| 75 | 23.42 | 23.58 | 23.56 | 23.59 | 23.62 | **23.77** | 23.74 |
| | 0.7011 | 0.7102 | 0.7109 | 0.7117 | 0.7114 | **0.7125** | 0.7122 |
| 100 | 22.13 | 22.16 | 22.25 | 22.28 | 22.27 | 22.39 | **22.43** |
| | 0.6258 | 0.6251 | 0.6266 | 0.6270 | 0.6262 | **0.6283** | 0.6279 |

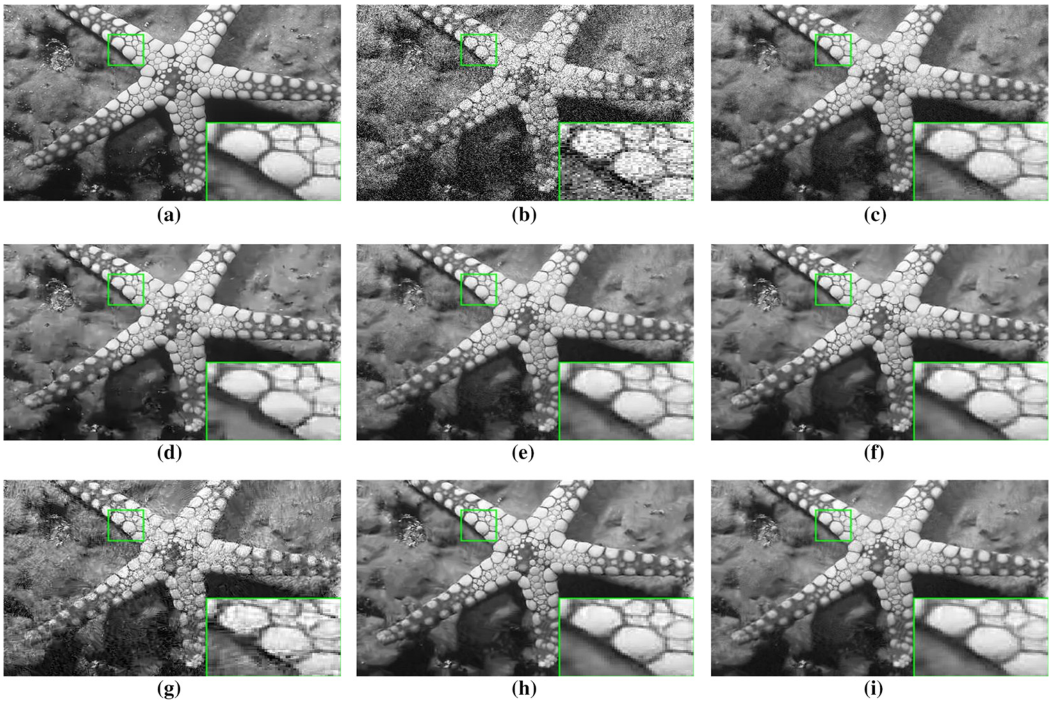

**Fig. 11** Denoising results on image “12003” from BSD by different methods ($\sigma = 25$): (a) ground truth (PSNR/SSIM), (b) noisy image (20.18/0.4151) (c) BM3D (27.36/0.8179), (d) EPPL (27.32/0.8154), (e) FRIST (27.60/0.8192), (f) WNNM (27.57/0.8167), (g) STROLLR (27.49/0.8172), (h) PGPD (27.53/0.8161), and (i) EPL-IMSC (27.76/0.8186).

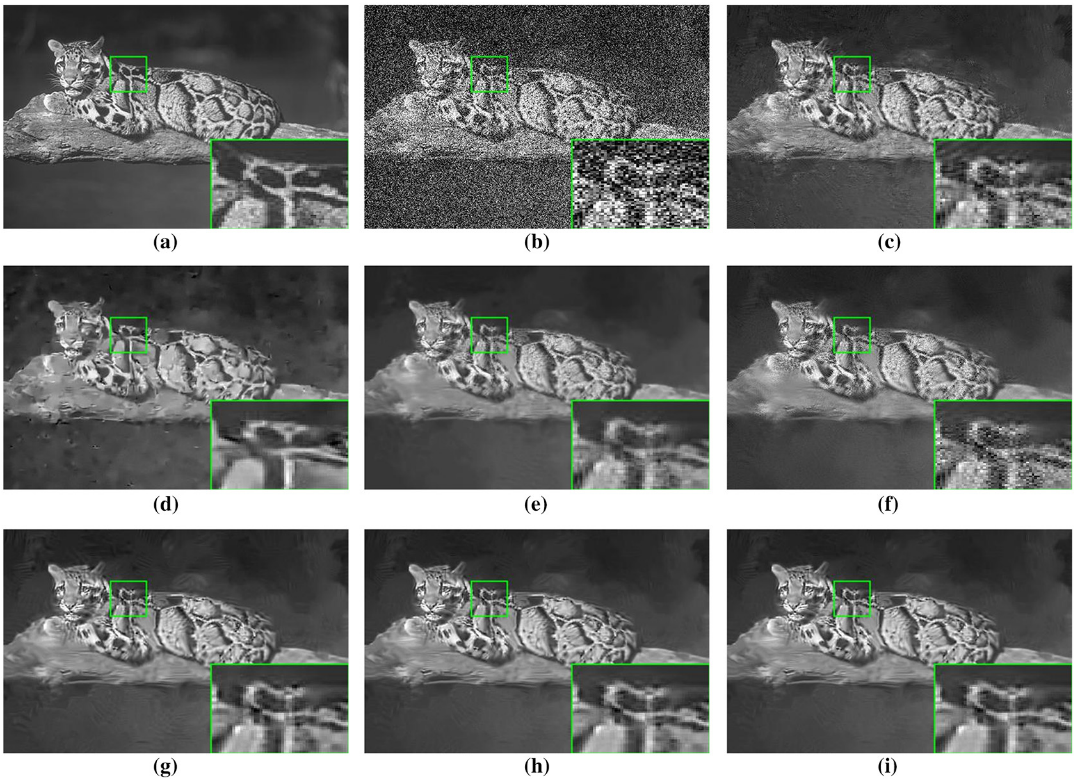

**Fig. 12** Denoising results on image “160068” from BSD by different methods ($\sigma = 50$): (a) ground truth (PSNR/SSIM), (b) noisy image (14.16/0.1730) (c) BM3D (24.91/0.8091), (d) EPPL (24.83/0.7939), (e) FRIST (25.07/0.8051), (f) WNNM (25.11/0.7886), (g) STROLLR (25.02/0.7964), (h) PGPD (25.26/0.8093), and (i) EPL-IMSC (25.13/0.8090).

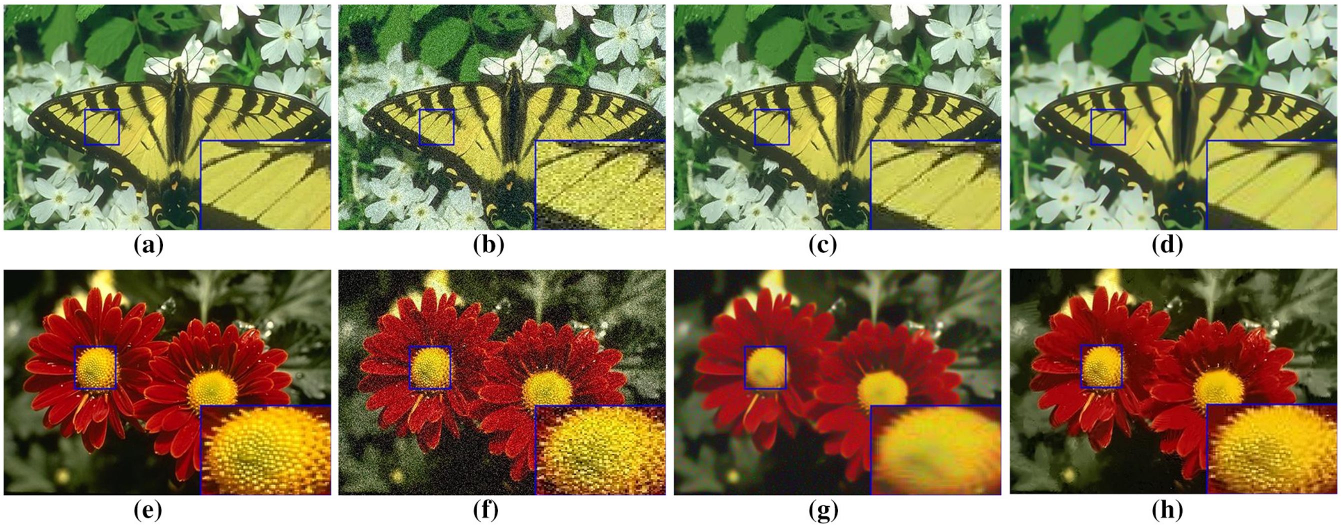

**Fig. 13** Denoising results on color images “35010” and “124084,” respectively. (a)–(d) AWGN ($\sigma = 20$): (a) ground truth (PSNR/SSIM), (b) noisy image (23.11/0.5778) (c) CBM3D (29.85/0.8725), and (d) EPPL-IMSC (30.05/0.8313); (e)–(h) AWGN ($\sigma = 40$): (e) ground truth (PSNR/SSIM), (f) noisy image (21.48/0.4132) (g) CBM3D (28.22/0.7827), and (h) EPPL-IMSC (28.30/0.7835).

**Table 5** Average run time (s) comparison on images of size 256 × 256 with different noise levels

| $\sigma$ | BM3D | EPLL | FRIST | WNNM | STROLLR | PGPD | EPL-IMSC |
|---|---|---|---|---|---|---|---|
| 25 | 0.70 ± 0.09 | 38.52 ± 0.07 | 355.24 ± 5.38 | 74.77 ± 6.32 | 289 ± 8.73 | 9.12 ± 1.03 | 150.73 ± 7.12 |
| 50 | 0.87 ± 0.05 | 40.21 ± 1.83 | 326.71 ± 9.21 | 116.83 ± 7.65 | 329 ± 10.51 | 10.47 ± 3.17 | 228.39 ± 5.63 |
| 75 | 0.89 ± 0.06 | 40.91 ± 1.33 | 369.86 ± 7.80 | 177.37 ± 8.49 | 355 ± 13.42 | 12.81 ± 1.22 | 251.89 ± 9.62 |
| 100 | 0.91 ± 0.04 | 42.80 ± 1.96 | 384.59 ± 13.79 | 185.21 ± 11.91 | 367 ± 11.41 | 13.23 ± 2.61 | 266.15 ± 12.18 |

parameters under the Bayesian framework. Finally, an iterative shrinkage algorithm was applied to solve the $l_1$-optimization problem. Extensive experiments indicated that the EPL-IMSC model not only achieves good denoising effect but also preserves the local details of the image very well.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61877038), the Science Research and Development Program of Shaanxi Province of China (No. 2016NY-176), and the Natural Science Basic Research Plan in Shaanxi Province of China (No. 2018JM6068).

## References

1. M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Trans. Signal Process.* **54**(11), 4311–4322 (2006).
2. J. Xu, L. Zhang, and D. Zhang, “A trilateral weighted sparse coding scheme for real-world image denoising,” in *Eur. Conf. Comput. Vision*, Vol. **11212**, pp. 21–38 (2018).
3. L. Wang, K. Lu, and P. Liu, “Compressed sensing of a remote sensing image based on the priors of the reference image,” *IEEE Geosci. Remote Sens. Lett.* **12**(4), 736–740 (2015).
4. J. Xu, L. Zhang, and D. Zhang, “External prior guided internal prior learning for real-world noisy image denoising,” *IEEE Trans. Image Process.* **27**(6), 2996–3010 (2018).
5. L. Liu et al., “Weighted joint sparse representation for removing mixed noise in image,” *IEEE Trans. Cybern.* **47**(3), 600–611 (2017).
6. B. Wen, S. Ravishankar, and Y. Bresler, “FRIST—flipping and rotation invariant sparsifying transform learning and applications,” *Inverse Prob.* **33**(7), 074007 (2017).
7. C. Knaus and M. Zwicker, “Progressive image denoising,” *IEEE Trans. Image Process.* **23**(7), 3114–3125 (2014).
8. L. Zhao et al., “Local activity-driven structural-preserving filtering for noise removal and image smoothing,” *Signal Process.* **157**, 62–72 (2019).
9. S. G. Chang, B. Yu, and M. Vetterli, “Adaptive wavelet thresholding for image denoising and compression,” *IEEE Trans. Image Process.* **9**(9), 1532–1546 (2000).
10. J. Starck, D. L. Donoho, and E. J. Candes, “The curvelet transform for image denoising,” *IEEE Trans. Image Process.* **11**(6), 670–684 (2002).
11. H. Wang et al., “Low-rank matrix recovery via smooth rank function and its application in image restoration,” *Int. J. Mach. Learn. Cybern.* **9**, 1565–1576 (2018).
12. H. Wang et al., “Reweighted low-rank matrix analysis with structural smoothness for image denoising,” *IEEE Trans. Image Process.* **27**(4), 1777–1792 (2018).
13. X. Jia, X. Feng, and W. Wang, “Rank constrained nuclear norm minimization with application to image denoising,” *Signal Process.* **129**, 1–11 (2016).
14. J. Xue et al., “Joint spatial and spectral low-rank regularization for hyperspectral image denoising,” *IEEE Trans. Geosci. Remote Sens.* **56**(4), 1940–1958 (2018).
15. T. Huang et al., “Mixed noise removal via Laplacian scale mixture modeling and nonlocal low-rank approximation,” *IEEE Trans. Image Process.* **26**(7), 3171–3186 (2017).
16. J. Xu et al., “Multi-channel weighted nuclear norm minimization for real color image denoising,” in *IEEE Int. Conf. Comput. Vision (ICCV)*, IEEE, pp. 1105–1113 (2017).

17. B. Wen, Y. Li, and Y. Bresler, "When sparsity meets low-rankness: transform learning with non-local low-rank constraint for image restoration," in *IEEE Int. Conf. Acoust., Speech and Signal Process. (ICASSP)*, IEEE, pp. 2297–2301 (2017).
18. A. Parekh and I. W. Selesnick, "Enhanced low-rank matrix approximation," *IEEE Signal Process. Lett.* **23**(4), 493–497 (2016).
19. S. Osher et al., "An iterative regularization method for total variation based image restoration," *Multiscale Model. Simul.* **4**(2), 460–489 (2005).
20. Y. Weiss and W. T. Freeman, "What makes a good model of natural images?" in *IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, IEEE, pp. 1–8 (2007).
21. K. Zhang et al., "Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.* **26**(7), 3142–3155 (2017).
22. K. Zhang, W. Zuo, and L. Zhang, "FFDNet: toward a fast and flexible solution for CNN-based image denoising," *IEEE Trans. Image Process.* **27**(9), 4608–4622 (2018).
23. S. Roth and M. J. Black, "Fields of experts," *Int. J. Comput. Vision* **82**(2), 205–229 (2009).
24. X. Lan et al., "Efficient belief propagation with learned higher-order Markov random fields," *Lect. Notes Comput. Sci.* **3952**, 269–282 (2006).
25. K. Dabov et al., "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.* **16**(8), 2080–2095 (2007).
26. J. Portilla et al., "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Trans. Image Process.* **12**(11), 1338–1351 (2003).
27. S. Gu et al., "Weighted nuclear norm minimization and its applications to low level vision," *Int. J. Comput. Vision* **121**(2), 183–208 (2017).
28. J. Jiang, L. Zhang, and J. Yang, "Mixed noise removal by weighted encoding with sparse nonlocal regularization," *IEEE Trans. Image Process.* **23**(6), 2651–2662 (2014).
29. J. Xu et al., "Patch group based nonlocal self-similarity prior learning for image denoising," in *IEEE Int. Conf. Comput. Vision (ICCV)*, IEEE, pp. 244–252 (2015).
30. S. Yao et al., "Principal component dictionary-based patch grouping for image denoising," *J. Visual Commun. Image Represent.* **50**, 111–122 (2018).
31. P. Liu et al., "Remote-sensing image denoising using partial differential equations and auxiliary images as priors," *IEEE Geosci. Remote Sens. Lett.* **9**(3), 358–362 (2012).
32. A. Buades, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising," *IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit. (CVPR)*, Vol. 2, pp. 60–65 (2005).
33. Z. Gao et al., "Fast sparse coding for range data denoising with sparse ridges constraint," *Sensors* **18**(5), 1449 (2018).
34. Y. Zhang et al., "Low-rank graph regularized sparse coding," *Lect. Notes Comput. Sci.* **11012**, 177–190 (2018).
35. Z. Shu, X. Wu, and C. Hu, "Structure preserving sparse coding for data representation," *Neural Process Lett.* **48**, 1705–1719 (2018).
36. Kodak, "Kodak lossless true color image suite," 2013, http://r0k.us/graphics/kodak/.
37. V. Papyan and M. Elad, "Multi-scale patch-based image restoration," *IEEE Trans. Image Process.* **25**(1), 249–261 (2016).
38. D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Int. Conf. Comput. Vision (ICCV)*, pp. 479–486 (2011).
39. A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc. Ser. B* **39**(1), 1–22 (1977).
40. B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature* **381**(6583), 607–609 (1996).
41. M. Zibulevsky and M. Elad, "L1–L2 optimization in signal and image processing," *IEEE Softw.* **27**(3), 76–88 (2010).
42. I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.* **57**(11), 1413–1457 (2004).
43. X. Zhang et al., "Bregmanized nonlocal regularization for deconvolution and sparse reconstruction," *SIAM J. Imaging Sci.* **3**(3), 253–276 (2010).
44. I. Selesnick, "Sparse regularization via convex analysis," *IEEE Trans. Signal Process.* **65**(17), 4481–4494 (2017).
45. L. Şendur and I. W. Selesnick, "A bivariate shrinkage function for wavelet-based denoising," in *IEEE Int. Conf. Acoust., Speech, and Signal Process. (ICASSP)*, IEEE, pp. II-1261–II-1264 (2002).
46. M. Elad, *Sparse and Redundant Representations*, Springer, New York (2010).
47. A. Lanza et al., "Nonconvex nonsmooth optimization via convex-nonconvex majorization-minimization," *Numerische Math.* **136**(2), 343–381 (2017).
48. P. Arbelaez et al., "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(5), 898–916 (2011).
49. K. Dabov et al., "Color image denoising via sparse 3-D collaborative filtering with grouping constraint in luminance-chrominance space," in *IEEE Int. Conf. Image Process. (ICIP)*, IEEE, pp. I-313–I-316 (2007).

**Qiongshuai Lyu** received his BS degree from the School of Computer at Henan University of Urban Construction and his MS degree from the School of Information Engineering at Zhengzhou University in 2007 and 2011, respectively. Now, he is a PhD candidate at the School of Computer Science, Shaanxi Normal University, China. His main research interests include image restoration, subspace clustering, and sparse signal representation models.

**Min Guo** received her MS and PhD degrees in the electronic technology application from the College of Physics and Information Technology, Shaanxi Normal University, Xi'an, China, in 1990 and 2003, respectively. She worked at the Postdoctoral Research Station, Northwestern Polytechnical University, Xi'an, China, in 2007. Since then, she has been a professor at the School of Computer Science, Shaanxi Normal University, Xi'an, China. Her research interests include pattern recognition, computer vision, and signal and image processing.

**Miao Ma** received her MS degree from the College of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an, China and her PhD from the School of Computer Science, Northwestern Polytechnical University, Xi'an, China, in 2002 and 2005, respectively. She is now a professor at the School of Computer Science of Shaanxi Normal University. Her main research interests include image processing, machine learning, and sparse signal representation.

**Richard Mankin** received his MS and PhD degrees from the University of Florida, Florida, USA, in 1976 and 1979, respectively. Now, he works at the Agricultural Research Service, United States Department of Agriculture, Center for Medical, Agricultural, and Veterinary Entomology, Gainesville, Florida, USA. His research includes digital signal processing, biological control, and acoustic detection of hidden insect infestations.