

RESEARCH

Open Access



# Image denoising based on global image similar patches searching and HOSVD to patches tensor

Jiye Guo<sup>1</sup>, Huayan Chen<sup>1</sup>, Zhengwei Shen<sup>1\*</sup>  and Ziqing Wang<sup>2</sup>

\*Correspondence:

ijwmip@ustb.edu.cn

<sup>1</sup> School of Mathematics and Physics, University of Science and Technology Beijing, Beijing 100083, China  
Full list of author information is available at the end of the article

## Abstract

Nonlocal self-similarity of natural image is an essential property for image processing. But how to measure the similarity between different patches and how to better exploit the similarity of patches are two crucial problems for image denoising. In this article, we establish a novel image denoising method based on a global image similar patches searching method and the similar patches tensor high-order singular value decomposition theory. Particularly, in order to find the reasonable similar patches to a reference, a variant of Gaussian mixture model (GMM) global similar patches searching method is proposed, and the graph process unit-based GMM model training method is performed to speed up the training process. Furthermore, the  $k$ -means and the local inter-class searching are used to improve on the similarity measure between patches. To better exploit the similarity of patches to image denoising, we rearrange similar patches to a tensor for each reference, and an iterative adaptive weighted tensor low-rank approximation method is established to perform image denoising. Experimental results clearly show that the proposed method is comparable to many recent denoising algorithms from the PNSR, SSIM and NCC viewpoint. For higher noise level (e.g.,  $\sigma = 50$ ), the performance of our proposed algorithm evinces a 0.06-unit, 0.05-unit and 0.12-unit improvement in the PNSR, SSIM and NCC score, respectively, when compared to state-of-the-art.

**Keywords:** Similarity patches searching, GMM, Iterative weights low-rank tensor approximation, Patches tensor

## 1 Introduction

As a fundamental topic in image and video processing, denoising has been widely studied in [1–6]. In recent years, image denoising methods mainly exploit three paradigms, i.e., global methods, filtering methods and patch-based methods. Originated from the literature [7], patch-based approaches [8–12] have shown great success to various image processing tasks, in which the essence of these methods is to exploit the recurrence of similar patches extracted from the noise image and the nonlocal self-similarity of natural images.

The most important thing of these patch-based methods is how to accurately measure the similarity between different patches of a natural image. Nearest neighbor search

(NNS) that based on Euclidean distance (ED), that is the square error of corresponding two patches, is one of widely used patches searching method, and the so-called BM3D algorithm [13] is one of its classical application. NNS method performs similar patches searching in a local window of image, thus is a local searching method. However, patches with significant structures, for example round edges or corners in an image, do not have a repeating pattern within a local window, which indicates that only using NNS to find similar patches is not an optimal method.

In order to better capture the structural similarity patches in natural images, various similar patches searching methods have been established in the literature. In [14], a so-called *Needle* patch representation method is proposed to improve on the reliability of patch-matching when the image quality deteriorates, and the patch representation *Needle* consists of small multi-scale versions of the patch and its immediate surrounding region. With the development of recent innovations in training deep convolutional neural network, a deep learning local image patches matching methods that based on Triplet and Siamese networks using a combination of triplet loss and global loss has been proposed in [15]. In [16], a 2-channel based neural network is established to learn a general image patches similarity function directly from raw image pixels.

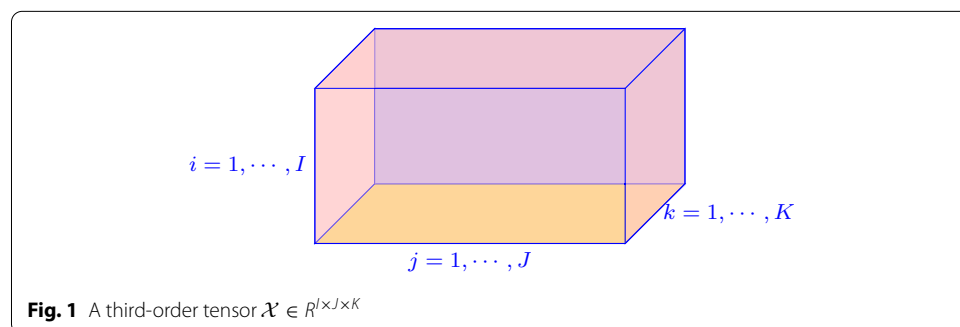
Patch priors that give high likelihoods for patches will yield better patches restoration performance [17]. Recently, using learned specific probabilistic distribution of patches from noise-free image dataset to globally search and recover similar patches from a degraded image has become a hot research issue in image processing. In principle, the probabilistic distribution of patches can be modeled using arbitrary distributions, but most commonly, Gaussian mixture model (GMM) is often used to model patches priors [17], even though the literature [18] argues that a generalized Gaussian mixture model is a better fit for image patches prior modeling. In fact, patches priors are learned over small image patches by GMM will make computational tasks such as learning, inference and likelihood estimation much easier than working with whole images. GMM also has been shown to be a powerful tool for patches classification and for similar patches matching. In addition, the learned means, covariance matrices and mixing weights over all patches can greatly improve the accuracy for searching similar patches.

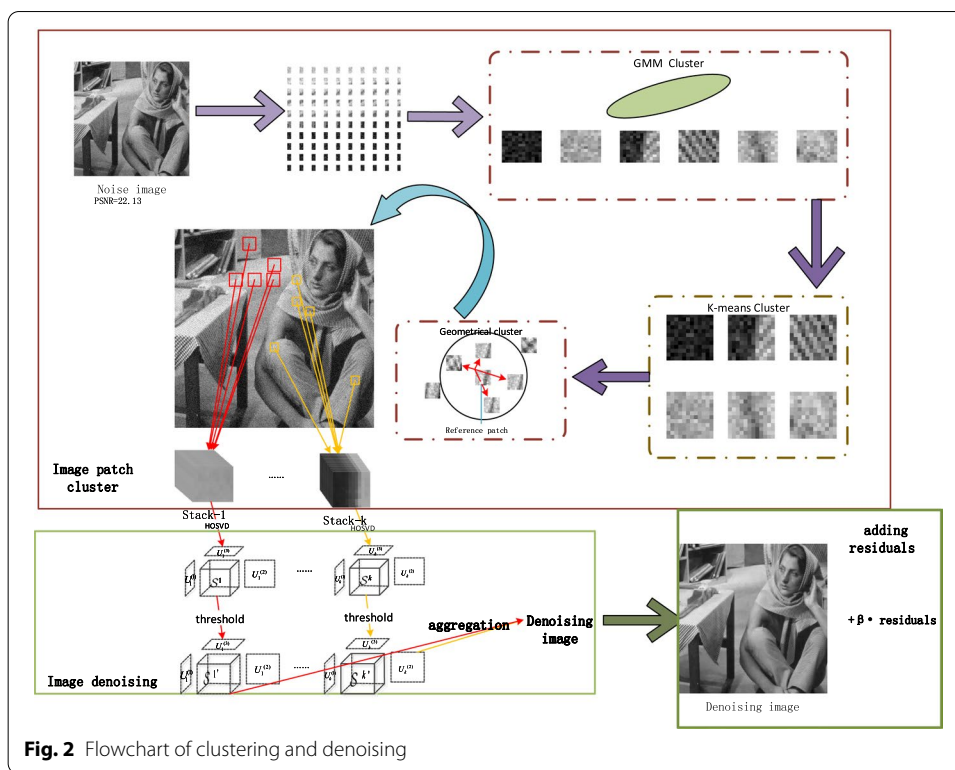
The expected patch log likelihood (EPLL) algorithm [17] based on GMM employs a global prior to search patch such that every selected patch is more like the given local prior. In [19], by integrating the external patch prior and internal self-similarity into one framework, the learned GMM (external patch prior) is used to guide the clustering of patches from the input degraded images, and then a patches matrix low-rank approximation process (internal self-similarity) is used to estimate the latent subspace for image recovery. In [20], a patch searching method that clusters similar patch candidates into patch groups using GMM-based clustering is proposed, and the selected patch groups that contain the reference patch are used to image denoising.

However, the simple GMM clustering method generally puts some nonsimilar patches, especially geometric nonsimilar patches, into a group which indicates this kind of similarity measure between patches is not accuracy. In other words, we need a more refined clustering method to determine which patches are similar or nonsimilar, which will motivate us to refine the GMM patches clustering method. To this end, in this paper, we exploit two additional operations to refine the GMM patches cluster: firstly,

we employ a simple  $K$ -means methods to refine the patches classification by using the mean intensity of each patch; secondly, for each reference patch in a certain group, we gradually expand the search radius until the number of similar patches contained in this circle satisfy our needs. The aim to do this is because the more similar images patches are grouped together (in the patches matrix sense [19] or the patches tensor used in this paper), the better the patches denoising effect will be. To better exploit the spatial geometric structural information of similar patches, we apply the 3D patches tensor  $\mathcal{X}$ , as shown in Fig. 1, to organize the similar patches instead of vectorizing each patches to form a similar patches matrix as in [19], then low-rank tensor approximation method based on HOSVD is used to perform image denoising. An iterative adaptive weighted core tensor thresholding algorithm is proposed to achieve low-rank tensor approximation. The reason using this algorithm are twofold, firstly, the tensor nuclear norm [21] (in fact is the  $\ell_1$  norm penalization) will result in significantly biased estimation to the values in core tensor, thus cannot achieve a reliable image recovery. Secondly, although the nonconvex penalty, such as  $\ell_p$  with  $(0 \leq p < 1)$ , can also provide unbiased solution, it is computationally cumbersome. As a surrogate to nonconvex penalty [22], the weighted  $\ell_1$  norm not only can ameliorate the bias to the larger values in core tensor, but also is low computational complexity.

The initial motivation of this work come from two observations: firstly, tensor organization of image patches can preserve more structural information of image than matrix form of vectorized image patches exploited in the literature; secondly, the more accurate the distance/similarity measure between image patches is, the better the image denoising effect achieved by low-rank image patches tensor approximation method will be. These observations inspire us to propose an image similar patches clustering method, denoted as GKGLM, i.e., “GMM” + “ $k$ -means” + “inter-class geometry location similarity patches searching.” The advantages of these variants of GMM are the hybrids globally patches searching (i.e., GMM) and the locally patches searching (i.e., the  $k$ -means and inter-class geometry location similarity patches searching). Furthermore, consider the high computational complexity of training process of GMM using expectation maximization (EM) method, we propose a novel GPU based GMM training method using the back propagation and gradient descent method. Meanwhile, an iterative weighted image patches tensor approximation is proposed to perform image denoising, which the key advantage of this iterative method is the unbiased estimation to larger entry values in core tensor that are the main feature of images. See Fig. 2 for the flowchart of this paper.





The main contributions of this paper arise from the following analysis:

1. A global similar patches searching method is proposed based on GMM patches clustering method, the k-means method and a local searching method within patches group are also being integrated into this framework to improve the accuracy of this patches clustering method, which will preserve more geometric structural similar information of patches.
2. A GPU based parallel training method is proposed to accelerate the GMM modeling training solved by EM algorithm.
3. In order to preserve more spatial geometric structural image information and fine details information image during denoising, an iterative adaptive weighted patches tensor approximation method is proposed.

The rest of the paper is organized as follows. In Sect. 2, we provide a brief introduction of the GMM and discuss how to search similar patches of the reference patch, including the algorithm of training GMM by the expectation maximization (EM) algorithm. In Sect. 3, we introduce the definition of three-order tensor, the properties of the corresponding core tensor and HOSVD algorithm. The iterative weighted image patches tensor approximation algorithm for image denoising is discussed in Sect. 4. We present numerical experiments and results in Sect. 5. The conclusion is presented in Sect. 6.

## 2 Similar patches matching by deep cluster based GMM training

### 2.1 GMM patches prior modeling

As being beforementioned, the probabilistic distribution of patches can be modeled using arbitrary distributions; but most commonly, GMM is used to learn patches priors. We take advantage of GMM as a probabilistic model to learn patch priors from noise-free patches set; then, noisy image patches clustering guided by the learned Gaussian mixture priors. GMM makes global searching for similar patches of a reference patch in the whole noise image.

The probability of a random patch from clean images sets will be defined as follows:

$$p(x_i|\Theta) = \sum_{k=1}^K w_k p(x_i|\theta_k) \quad (1)$$

where  $\Theta = (w_1, \dots, w_K, \theta_1, \dots, \theta_K)$  is a set of parameters;  $w_k$  denotes the weight that probability function  $p(x_i|\theta_k)$  ( $k = 1, \dots, K$ ) contributes to  $p(x_i|\Theta)$  and  $\sum_{k=1}^K w_k = 1$ ;  $\theta_k = (\mu_k, \Sigma_k)$  are mean value and covariance matrix, respectively, in density function  $p(x_i|\theta_k)$ . One of the most important feature of the patches is that it is close to a low-dimensional manifold for many natural images. Therefore, to more accurately measure the distance between different patches, we replace the frequently used Euclidean distance (ED) with Mahalanobis distance (MD) to measure the patches difference used in patches cluster problem, i.e.,  $p(x_i|\theta_k) = c \cdot \exp(-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1}(x_i - \mu_k))$ , where  $c$  is a constant. As shown the patches correct matching rates results in Table 1 from [19], even though the correct matching rates of ED is better than that of MD in image homogenous region, the MD overwhelmingly surpass that of ED in the structural and textural region, which indicates that using Mahalanobis distance to measure the difference between image patches is more reasonable than using Euclidean distance.

### 2.2 Parameters estimation using EM algorithm based on GPU

To accelerate image denoising algorithm, we exploit a pre-trained GMM for the first step to cluster the similar patches. That is the parameters in GMMs modeling in (1) will be learned from training image patches set using the expectation maximization (EM) algorithm. However, most traditional implementation of EM algorithm is high computational complexity, thus cost a large amount of time in training processing. On the other hand, because the most time-consuming process in training is the calculation of distance between different patches, thus, it is easy to consider to apply parallel computing to accelerate GMM training with GPU.

**Table 1** Patches correct matching rates (ED vs. MD) [19]

$\sigma_{\text{noise}}$	Smooth		Structural		Textural	
	MD	ED	MD	ED	MD	ED
$\sigma = 35$	6.2	<b>8.1</b>	<b>36.4</b>	23.0	<b>65.6</b>	57.3
$\sigma = 55$	4.2	<b>7.2</b>	<b>24.5</b>	13.4	<b>53.1</b>	36.5
$\sigma = 75$	3.4	<b>6.9</b>	<b>17.0</b>	10.0	<b>43.0</b>	24.3

Bold value indicates the best results compared between MD and ED

It is well known that an EM based GMM training methods include these two steps, i.e., expectation step and maximization step, respectively. Expectation step in EM algorithm is a kind of evaluation and cluster processing, that is the model parameters learned in the maximization step will be checked by evaluating the difference between the last and the current clustering results. And the maximization step is the parameters estimation processing, which is used to fit the new cluster results by using maximum likelihood estimation (MLE) method.

In the following, we establish a deep cluster based GMM training to accelerate the EM training algorithm. Particularly, in the parameters estimation step (maximization step), we first establish a binary cross-entropy (BCE) loss function based on the difference between the last clustering results  $Y^k$  and the current clustering results  $\hat{Y}$  in the testing step (i.e., the Expectation step), then, the back-propagation algorithm and the stochastic gradient descent (SGD) method based on GPU are used to estimate the parameters of GMM modeling. See more details in Algorithm (1). We train the GMM prior model from a set of 2 million overlapping patches that randomly sampled from [23] with their DC removed. Allowing for overlapping when sampling patches is important because otherwise there would appear blocking artifact. See Fig. 3 for the GMM model learned flowchart.

---

**Algorithm 1** Deep Cluster Based GMM Training

---

**Input:** Patches from dataset:  $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p\}$ , Goal accuracy:  $\varepsilon$

**Output:** Trained GMM parameters  $\Theta$

- 1: **Initialization:** Find a group of parameters randomly as  $\Theta^{(0)} = \{\omega_1, \dots, \omega_K, \theta_1, \dots, \theta_K\}$
- 2: While ( $accuracy \leq \varepsilon$ )

- (i) (TESTING) Classify  $X$  with  $f_{\Theta^{(m-1)}}(\mathbf{x}_i)$  as  $Z^{(m)}$  where

$$z_i^{(m)} = f_{\Theta^{(m-1)}}(\mathbf{x}_i) = \arg \max_j \omega_j p(\mathbf{x}_i | \theta_j^{(m-1)}) \tag{2}$$

build the current distribution  $Y^{(m)}$  s.t.

$$y_{ij}^{(m)} = \begin{cases} 1 & \text{if } j = z_i \\ 0 & \text{if } j \neq z_i \end{cases} \tag{3}$$

calculate the accuracy and check the terminated condition

$$accuracy = \frac{1}{p} \sum_{i=1}^p \sum_{j=1}^p y_{ij}^{(m-1)} y_{ij}^{(m)} \tag{4}$$

- (ii) (TRAINING) Using BP and SGD, find the  $\Theta$  s.t.

$$\begin{aligned} \Theta^{(m)} &= \arg \min_{\Theta} BCE(Y^{(m)}, \hat{Y}) \\ &= \arg \max_{\Theta} \sum_{i=1}^p \log p_{z_i^{(m)}}(\mathbf{x}_i | \Theta) \\ &= \arg \max_{\theta, \omega} \sum_{i=1}^p \log \omega_{z_i^{(m)}} p(\mathbf{x}_i | \theta_{z_i^{(m)}}) \end{aligned} \tag{5}$$


---

### 2.3 Similar patches searching by deep cluster based GMM training

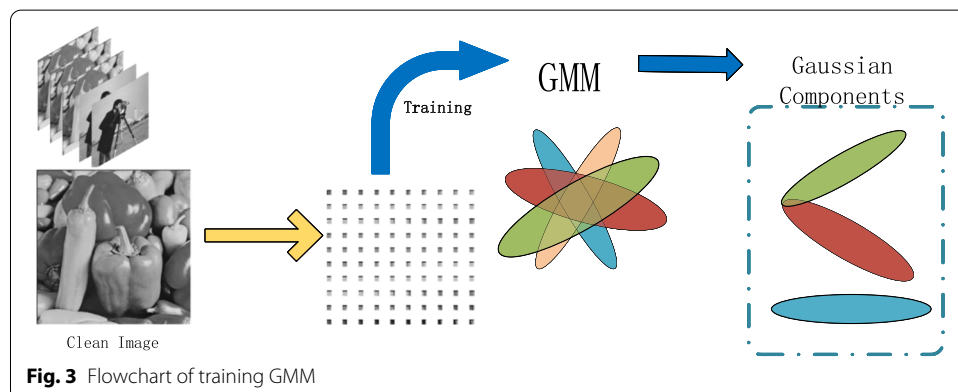
Based on the previously trained GMM patches model, given a reference patch, we propose a “GMM” + “*k*-means” + “inter-class geometry location similarity patches searching” method to refine the patches similarity matching.

Given a noisy image *Y*, we firstly split it into a set of overlapping patches, then we denote  $P_i Y$  by the *i*-th patch of *Y*. The likelihood of the *i*-th patch belongs to the *k*th ( $k = 1, \dots, K$ ) cluster under parameter  $\Theta$  is defined:

$$p(k|P_i Y) = \frac{w_k p(P_i Y|\theta_k)}{\sum_{j=1}^K w_j p(P_i Y|\theta_j)}. \tag{6}$$

It should be noticed that here  $\Theta = (w_1, \dots, w_K, \theta_1', \dots, \theta_K')$  with  $\theta_i = (\mu_i, \Sigma_i')$  are estimated by  $\Sigma_k' = \Sigma_k + \sigma^2 I$ , where  $\Sigma_k$  is a covariance matrix of the learned GMMs,  $\sigma$  is the noise standard deviation and *I* is an identity matrix. Maximizing the likelihood (6) to determine Gaussian component that makes the likelihood maximum, then we denote it to be the  $\hat{k}$ th Gaussian component and assign the *i*th noisy patch into the  $\hat{k}$  cluster. This is the GMM based noisy patches globally cluster, and all patches of the noisy image *Y* are split into different several clusters. As we have been emphasized in Sect. 2.1, there exists a deficiency of GMM based clustering method, i.e., in image homogenous region, the Euclidean distance based clustering method is better than the GMM based patches clustering method. Motivated by this observation, to improve the accuracy of patches matching, we employ a simple *K*-means method to each GMM based cluster to refine patches classification. After that, if there exists a class with the number of patches in it is less than 10, we will merge the patches in this class to the other classes most likely.

Next, we explain how to choose number of similar patches of a reference patch in each class to form a patches tensor. We firstly determine which class each reference patch is in, then we select similar patches for each reference from this class by calculating the location distance from reference patch to each patch in this class, i.e., the distance from the upper-left pixel of reference patch and the patches in this class. Secondly, the first *T* least distance corresponding to the patches is selected to form a similar patches tensor. We call this process to be the inter-class geometry location similarity patches searching method.



Thus, our similarity patches searching method can utilize the global information and the local location information of image. For convenience, we denote this clustering method as GKGLM, i.e., “GMM” + “*k*-means” + “inter-class geometry location similarity patches searching.” At last, we summarize the GKGLM algorithm in Algorithm 2.

---

**Algorithm 2** GKGLM for patch clustering

---

**Input:** Noisy image  $Y$ , the learned GMM model parameter  $\Theta$ , the number of reference patches  $S$

**Output:** Image patch tensor  $\{\mathcal{X}^s\}_{s=1:S}$  Iteration  $s = 1 : S$

- (i) Calculate likelihood  $p(s|P_i Y)$  via. Eq 6;
  - (ii) *K*-means cluster in GMM based class;
  - (iii) Inter-class geometry location similarity patches searching;
  - (iv) Create  $\mathcal{X}^s$  by stacking the selected patches corresponding to the reference patch.
- 

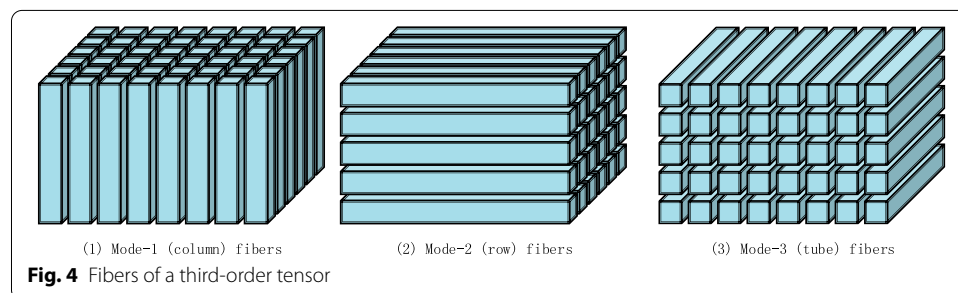
### 3 Low-rank image patches tensor approximation

#### 3.1 Rank of patches tensor and HOSVD

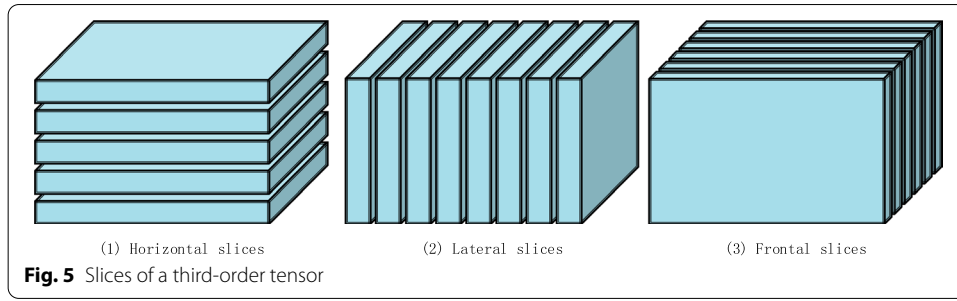
Third-order patches tensors have column, row and tube fibers; Figs. 4 and 5 show the horizontal, lateral and frontal slides of a third-order. Low-rank matrix approximation methods are to measure the low-rank structure of the observed matrix  $X$  by minimizing its rank. Similarly, the latent tensor data can be approximated from low-rank version of observed tensor measurements. But, the low-rank decomposition to a multidimensional tensor usually is ticklish, and generally, there exist at least three different rank definitions of multidimensional tensor relative to different low-rank tensor decompositions, and it is hard to find a tight convex relaxation to the nonconvex rank function of multidimensional tensor. Two often used rank definitions of tensor are the CANDECOMP/PARAFAC (CP) rank and Tucker rank.

**Definition 1** The CP rank of a tensor  $\mathcal{X} \in R^{I_1 \times \dots \times I_N}$  is defined as the minimum number of rank-1 decomposition,

$$\text{rank}_{CP}(\mathcal{X}) = \min \left\{ r \mid \mathcal{X} = \sum_{i=1}^r c_i \mathcal{V}_i \right\} \tag{7}$$







where  $\mathcal{V}_i = a_1^{(i)} \otimes \dots \otimes a_k^{(i)}$ , symbol  $\otimes$  denotes vector outerproduct,  $a_j^{(i)} \in R^{n_j}$  is a vector and  $c_i$  is the decomposition coefficient [24].

**Definition 2** Tucker rank of a tensor  $\mathcal{X} \in R^{I_1 \times \dots \times I_N}$  is denoted by  $\text{rank}_T(\mathcal{X})$  and is a  $N$ -dimensional vector expressed as follows:

$$\text{rank}_T(\mathcal{X}) = (\text{rank}(X_{(1)}), \dots, \text{rank}(X_{(N)})) \tag{8}$$

where  $\text{rank}(X_{(i)})$  denotes the matrix rank of  $i$ th-mode matricization(see the following section)  $X_i$  to tensor  $\mathcal{X}$  for  $i = 1, 2, \dots, N$ .

Based on these two definitions of tensor rank, tensor has a similar nuclear norm like the matrix nuclear norm, i.e., tensor nuclear norm (TNN), which is one extension of matrix nuclear norm [21].

$$\|\mathcal{X}\|_* = \min_{\mathcal{X}} \sum_{i=1}^N \|X_{(i)}\|_* \tag{9}$$

The tensor trace norm (TTN) is similar to TNN, it is developed in [25]

$$\|\mathcal{X}\|_* = \sum_{i=1}^N \alpha_i \|X_{(i)}\|_* \tag{10}$$

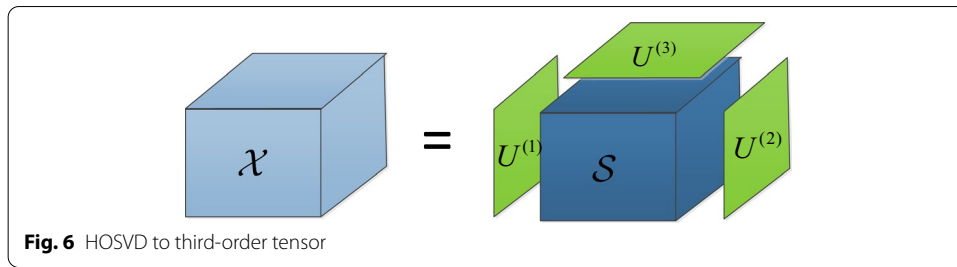
where  $\alpha_i > 0$  and  $\sum_{i=1}^N \alpha_i = 1$ .

TNN and TTN, in essence, merely is a convex combination of trace norms of the  $n$ th-mode unfolding  $X_{(i)}$ . A novel tensor nuclear norm  $\|\mathcal{X}\|_{\text{TNN}}$  is proposed in the literature [26] and is defined as the sum of the singular values of all frontal slices. This tensor nuclear norm has also been proved to be a norm and be the tightest convex relaxation to  $\ell_1$  norm of tensor multilinear rank.

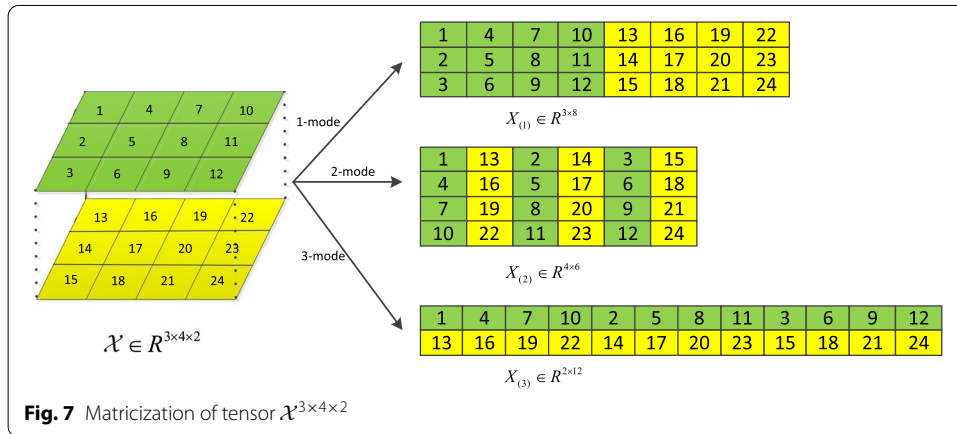
A HOSVD to a tensor  $\mathcal{X} \in R^{I_1 \times \dots \times I_N}$  can be expressed as a core tensor  $\mathcal{S} \in R^{r_1 \times \dots \times r_N}$  and matrices multiplication shown as follows,

$$\mathcal{X} = \mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 \dots \times_N U^{(N)} \tag{11}$$

where  $U^{(i)} \in R^{I_i \times r_i}$  ( $1 \leq i \leq N$ ) is the left singular orthogonal matrix calculated by performing SVD on matrix  $X_i$ . Figure 6 shows a HOSVD to a third-order tensor. The inverse HOSVD is defined by:



**Fig. 6** HOSVD to third-order tensor



**Fig. 7** Matricization of tensor  $\mathcal{X}^{3 \times 4 \times 2}$

$$\mathcal{S} = \mathcal{X} \times_1 U^{(1)T} \times_2 U^{(2)T} \times_3 \cdots \times_N U^{(N)T} \tag{12}$$

where  $(\cdot)^T$  denotes the matrix transpose operation.

In some tensor decompositions, we need to unfold or flatten the tensor into matrix, a.k.s. matricization. There are  $N$  mode matricization to a  $N$ th-order tensor. The  $n$ th-mode matricization of a tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$  is denoted by  $X_{(n)} \in \mathbb{R}^{I_n \times (I_1 I_2 \cdots I_{n-1} I_{n+1} \cdots I_N)}$  ( $n = 1, 2, \dots, N$ ) is a matrix. We also denote functions  $\text{unfold}_n(\cdot)$  and  $\text{fold}_n(\cdot)$  to be the  $n$ th-mode unfolding and folding operation, that is,  $X_{(n)} = \text{unfold}_n(\mathcal{X})$  and  $\mathcal{X} = \text{fold}_n(X_{(n)})$ .

There are many different arrangement ways of the  $n$ th-mode matricization, but in practice, permutation of elements doesn't affect the result of calculation [27]. See Fig. 7 for a third-order tensor matricization. The  $n$ th-mode product is denoted by  $\mathcal{X} \times_n U$  where  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$  is a tensor and  $U \in \mathbb{R}^{J \times I_n}$  is a matrix. We can obtain a tensor  $\mathcal{Y} = \mathcal{X} \times_n U$  of size  $I_1 \times I_2 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$ . The  $n$ th-mode product can be expressed by matrix multiplication using  $\text{fold}_n(\cdot)$  as follows

$$\mathcal{Y} = \text{fold}_n(U \times X_{(n)}) \Leftrightarrow Y_{(n)} = U \times X_{(n)} \tag{13}$$

At last, the Frobenius norm of tensor  $\mathcal{X}$  is denoted by  $\|\mathcal{X}\|_F = (\sum_{i_1, \dots, i_N} |x_{i_1, \dots, i_N}|^2)^{1/2}$ ; 2-norm of vector  $v$  is denoted by  $\|v\|_2$ .

### 3.2 Low-rank patches tensor approximation

In this section, we establish a low-rank patches tensor approximation method that begin by an example. Consider a tensor  $\mathcal{X} \in \mathbb{R}^{3 \times 2 \times 3}$  and the corresponding core tensor  $\mathcal{S}$  with three horizontal slices, shown in Fig. 8. Three associated singular values matrices  $\Sigma_{(i)}$ , ( $i = 1, 2, 3$ ) obtained by SVD to the corresponding  $n$ th-mode matricization of tensor  $\mathcal{X}$ , i.e.,  $X_{(i)}$ , ( $i = 1, 2, 3$ ), are listed as follows

$$\Sigma_{(1)} = \begin{bmatrix} 7.5284 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5.9433 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\Sigma_{(2)} = \begin{bmatrix} 9.3063 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.3221 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\Sigma_{(3)} = \begin{bmatrix} 7.2051 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5.9551 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.1502 & 0 & 0 & 0 \end{bmatrix}.$$

Three  $n$ th-mode matricization of core tensor  $\mathcal{S}$  shown as follows,

$$S_{(1)} = \begin{bmatrix} -7.1846 & 0.0615 & -0.1703 & 0.7654 & 0.0035 & 2.1071 \\ 0.2235 & -0.4906 & -5.9024 & -0.0991 & 0.2675 & 0.3350 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$S_{(2)} = \begin{bmatrix} -7.1846 & -0.1703 & 0.0035 & 0.2235 & -5.9024 & 0.2675 & 0 & 0 & 0 \\ 0.0615 & 0.7654 & 2.1071 & -0.4906 & -0.0991 & 0.3350 & 0 & 0 & 0 \end{bmatrix}$$

$$S_{(3)} = \begin{bmatrix} -7.1846 & 0.0615 & 0.2235 & -0.4906 & 0 & 0 \\ -0.1703 & 0.7654 & -5.9024 & -0.0991 & 0 & 0 \\ 0.0035 & 2.1071 & 0.2675 & 0.3350 & 0 & 0 \end{bmatrix}.$$

From this example, we can find that the quadratic sum of the  $i$ th row of  $S_{(n)}$  equals to the square of the  $i$ th singular values of the corresponding  $\Sigma_{(n)}$ , which proves one property, that is,  $\|S\|_F^2 = \|\mathcal{X}\|_F^2$ . For instance,  $7.2051^2 = 51.9134$  exactly equals  $(-7.1846)^2 + 0.0615^2 + 0.2235^2 + (-0.4906)^2$ . Not only that, but all the  $\|\Sigma_{(i)}\|_F^2$ , ( $i = 1, 2, 3$ ) are identical, and equal to  $\|S\|_F^2 = \|\mathcal{X}\|_F^2$ , i.e., 92. That is, we have illustrated the following properties:

$$\|S_{(n)}^l\|_2 = \sigma_{(n)}^l, (l = 1, \dots, I_n) \tag{14}$$

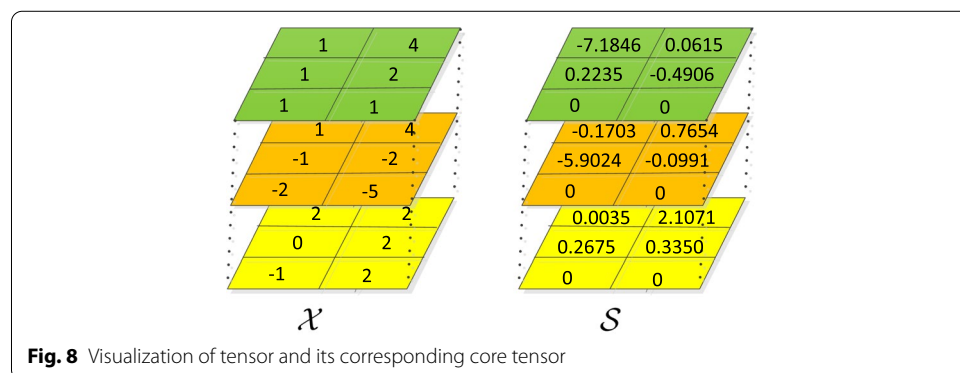


Fig. 8 Visualization of tensor and its corresponding core tensor

and

$$\|S\|_F^2 = \sum_{l=1}^{I_n} (\sigma_{(n)}^l)^2 = \|\mathcal{X}\|_F^2 \tag{15}$$

where  $S_{(n)}^l$  is the  $l$ th row vector of the  $n$ th-mode matricization of core tensor  $S$ ,  $\sigma_{(n)}^l$  denotes the  $l$ th singular value of the  $n$ th-mode matricization of tensor  $\mathcal{X}$ .

Based on these observations, we propose a novel low-rank patches tensor approximation method, i.e., by directly penalizing the adaptive weighted singular values of core tensor obtained by HOSVD to patches tensor.

#### 4 Image denoising algorithm based on low-rank patches tensor approximation

##### 4.1 Image denoising based on iterative low-rank patches tensors approximation algorithm

First, we introduce the objection function for image denoising in this paper.

$$\arg \min_X \frac{1}{2} \|Y - X\|_F^2 + \mathcal{W} \|F(X)\|_* \tag{16}$$

where  $Y$  and  $X$  denote noise image and the latent clean image, respectively,  $\mathcal{W}$  denotes adaptive weights, and  $\|F(X)\|_*$  denotes the proposed low-rank patches tensor nuclear norm (15).  $F(X)$  denotes tensor formulation and HOSVD to image  $X$ .

Low rank is a common assumption to natural image in image processing, so we usually use  $\ell_1$  or nuclear norm to approximate rank function.  $\ell_1$  or nuclear norm regularization can lead to sparse solution, and their convexity is another important reason why they have become so popular in signal processing field. However, both of them will result in significantly biased estimates since the convexity, thus, cannot achieve a reliable solution. In comparison, as a nonconvex penalty, such as the  $\ell_q$  ( $0 \leq q < 1$ ), smoothly clipped absolute deviation (SCAD) or minimax concave (MC) penalty is superior to  $\ell_1$  or nuclear norm regularization because it can ameliorate the bias problem. The advantages of nonconvex penalty regularization have been verified in many applications, and in fact, a nonconvex penalty can yield significantly better performance than a convex penalty regularization. On the other hand, nonconvex regularization based sparse and low-rank recovery are of becoming considerable interest in recent years, also partly attributes to the recent progress in nonconvex and nonsmooth optimization algorithm theory. Motivated by these observations, we exploit iterative adaptive weights to the regularization term. We claim that such an iterative adaptive weighted scheme is equivalent to a kind of nonconvex penalty to the core tensors, where the singular values are assigned different penalty weights.

Proximity operator plays a central role in tackling nonconvex and nonsmooth inverse problem, and it is a highly efficient first-order algorithms which can scale well for high-dimensional problems. For a proper and lower semi-continuous penalty function  $P_\lambda(\cdot)$ , where  $\lambda > 0$  is a threshold parameter, consider the following scalar proximal projection:

$$\text{proxp}_\lambda(t) = \arg \min_x \left\{ P_\lambda(x) + \frac{1}{2}(x - t)^2 \right\} \tag{17}$$

As  $P_\lambda(\cdot)$  is separable, the proximity operator of a vector  $\mathbf{t} = [t_1, \dots, t_n] \in R^n$ , denoted by  $\text{proxo}_\lambda(\mathbf{t})$  can be computed in an element-wise manner as:

$$\text{proxo}_\lambda(\mathbf{t}) = [\text{proxp}_\lambda(t_1), \dots, \text{proxp}_\lambda(t_n)]^T \tag{18}$$

For commonly used proximity operators, refer to Table 2.

For a patch tensor  $\mathcal{M}$ , we denote a generalized penalty for low-rank promotion to its core tensor by  $\hat{P}_\lambda(\cdot)$ , which is defined as :

$$\hat{P}_\lambda(\mathcal{M}) = \sum_i P_\lambda(\lambda_i) \tag{19}$$

where  $\lambda_i$  is the entry values in core tensor of tensor  $\mathcal{M}$ .

Exploiting the adaptive weights  $\ell_1$ -norm we can better preserve essential characteristic of image, and the nonconvex objective function can be solved easily by using the soft-thresholding operator. For each GKGLM patches tensor  $\mathcal{X}^k, k = 1 \dots K, K$  is the number of reference patches, we have the following minimization

$$\min_{\mathcal{X}} \mathcal{W}^k \|\mathcal{X}^k\|_* = \sum_j W_j^k |\lambda_j^k| \tag{20}$$

where  $W_j^k$  is the adaptive weight, and  $\lambda_j^k$  is the value in core tensor  $\mathcal{S}^k$  of tensor  $\mathcal{X}^k$ .

It is worth note that the adaptive weighted  $\ell_1$ -norm penalized optimization problem is nonconvex due to its varying weights  $W_j^k$ . But, if the weight  $W_j^k$  is nonincreasing assigned to the increasing core tensor absolute values  $|\lambda_j^k|$ , the penalized optimization problem is a convex function, the conclusion has been proved in the literature. In this paper, we are not to tackle the noise image directly, but to perform denoising for each tensor of being constituted by similar patches of reference patches. So, the objective function can be written as the following subproblems.

$$\arg \min_{\mathcal{X}_x^s} \frac{1}{2} \|\mathcal{X}_y^s - \mathcal{X}_x^s\|_F^2 + \sum_j W_j^s |\lambda_j^s|, \quad s = 1, \dots, S \tag{21}$$

where  $\mathcal{X}_y^s$  represents the  $k$ th reference patch tensor of noisy image  $Y$ ,  $\mathcal{X}_x^s$  is latent clear patches tensor corresponding to  $\mathcal{X}_y^s$ ,  $\lambda_j^s$  is the value of core tensor of noisy patches tensor  $\mathcal{X}_y^s$ , and  $W_j^s$  is the adaptive weight and assigned to  $|\lambda_j^s|$ . It is obvious that the optimization problem can be solved easily by using the soft-thresholding operator. After achieving

**Table 2** Regularization penalties and the corresponding proximity operator ( $\lambda > 0$ )

Penalty name	Penalty formulation	Proximity operator
Soft-thresholding	$P_\lambda(x) = \lambda x $	$\text{proxp}_\lambda(t) = \text{sign}(t) \max\{ t  - \lambda, 0\}$
Hard thresholding	$P_\lambda(x) = \lambda[2 - ( x  - \sqrt{2})^2 / ( x  < \sqrt{2})]$ or $P_\lambda(x) = \lambda x _0$	$\text{proxp}_\lambda(t) = \begin{cases} 0 &  t  < \sqrt{2}\lambda \\ \{0, t\} &  t  = \sqrt{2}\lambda \\ t &  t  > \sqrt{2}\lambda \end{cases}$

$\mathcal{X}_x^s$  for each given noisy tensor  $\mathcal{X}_y^s$ , we will put the new patches back into the image to constitute a new one, and an aggregation procedure is needed. Until now, we finish the  $n$ th image denoising and obtain the clean image  $\hat{X}_n$ .

#### 4.2 Adaptive weight setting to $W_j^s$

Adding residual  $(Y - \hat{X}_n)$  of  $n$ th iteration back to the  $(n + 1)$ th step denoised image  $Y_{n+1}$  is the essence of the iterative algorithm. For a better denoising performance, we formulate the  $Y_{n+1}$  as follows:

$$Y_{n+1} = \hat{X}_n^s + \eta(Y - \hat{X}_n^s) \tag{22}$$

where  $n$  is the outer iteration number,  $k$  is the inner iteration number, and  $\eta$  is the relaxation parameter. Since, we want to add the residual back to the denoised image, the variance of noise remaining can be estimated by

$$\sigma_n = \gamma \sqrt{\sigma^2 - \frac{1}{k \times l} \|Y - Y_{n-1}\|_F^2} \tag{23}$$

where  $\sigma$  is noise variance of  $Y$ ,  $\gamma$  is a scaling factor used to control the re-estimation of noise variance,  $k \times l$  is the number of all pixels in  $Y$ .

A advantages of adaptive-thresholding is to preserve large coefficients while filters small coefficients, so the features in image can be survived through thresholding. Then, we set the weight corresponding to each  $\lambda_j$  to be:

$$W_j^s = \frac{2\sqrt{2}\sqrt{N}\sigma_n^2}{|\lambda_j| + \varepsilon} \tag{24}$$

where  $N$  is the number of image patches of each tensor  $\mathcal{S}_y^s$ ,  $\sigma_n$  can be computed by 23,  $\varepsilon > 0$  is a sufficient small positive parameter for avoiding dividing by zero. We apply  $W_j^s$  as adaptive weight to solve the optimization problem 21 by the soft-thresholding proximal operator shown in Table 2, thus solution  $\tau_j^s$  to the  $j$ th element of  $\mathcal{S}_x^s$  can be expressed by as follows

$$\tau_j^k = \text{sign}(\lambda_j) \max\{|\lambda_j| - W_j^s, 0\}. \tag{25}$$

#### 4.3 Aggregation

When we put these patches back into the image, different patches tensor will be assigned different weights. Therefore, we use the weighted averaging to give more weight to patches tensors with less noise and less weight to those with much noise. Specifically, we use the following equation to define weight.

$$w = \begin{cases} \frac{p^2 \times N}{p^2 \times N + I}, & \text{if } I \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

where  $N$  is the number of patches in current tensor,  $p$  is the patch size and  $I$  is the number of thresholded elements in core tensor of this patches tensor. The result image can be calculated by the following formulation

$$\hat{X}_{(x,y)} = \frac{\sum_{i \in N_{x,y}} \sum_{j \in J(j)_{x,y}} w_{i,j} \Omega_{i,j}}{\sum_{i \in N_{x,y}} \sum_{j \in J(j)_{x,y}} w_{i,j}} \quad (27)$$

where  $N_{x,y}$  are all GKGLM patches tensor overlapping in position  $(x, y)$ ,  $J(i)_{x,y}$  are all patches in  $i$ th tensor that overlap in position  $(x, y)$  and  $\Omega_{i,j}$  is a pixel in the  $i$ th tensor, the  $j$ th tensor in position  $(x, y)$ . The proposed algorithm is summarized in Algorithm 3, we denote the algorithm as GKGLM-Tensor.

---

**Algorithm 3** GKGLM-Tensor image denoising algorithm

---

**Input:** Noisy image  $Y$ , noise standard deviation  $\sigma$ , the number of reference patches  $S$ , and iteration number  $L$

**Output:** denoised image  $X$

1: **Initialization:**

- 1) Choose a reasonable scaling factor  $\gamma$  for controlling the re-estimation of noise variance.
- 2) Initialize  $X_0 = Y$ ,  $Y^0 = Y$ ,  $\sigma_0 = \sigma$ , image patch tensor  $\mathcal{X}$

2: **Iteration:**

- 1) Outer loop : for  $n = 1 : L$  do
    - (i) Re-estimate  $Y^n$  by 22
    - (ii) Re-estimate noise variance  $\sigma_n$  by 23
  - 2) Inner loop : for  $s = 1 : S$  do
    - (i) Compute  $U^{(1)}$ ,  $U^{(2)}$ ,  $U^{(3)}$  and core tensor  $\mathcal{S}_y^{(s)}$  of  $\mathcal{X}_y^{(s)}$  by HOSVD via Eq 11
    - (ii) For each  $\lambda_j$  in core tensor  $\mathcal{S}_y^{(s)}$  calculate the  $W_j^s$  via Eq. 24
    - (iii) Apply threshold  $W_j^s$  to  $\lambda_j$  in  $\mathcal{S}_y^{(s)}$  via. Eq. 25
    - (iv) Estimate denoised patches tensor by 21.

End for
  - 3) Obtain the  $n$ th step denoised image  $\hat{X}_n$  vis Eq 27
 

End for
  - 4) Obtain the denoised image  $X$
- 

## 5 Experiments

In this section, we conduct image denoising experiments on several widely used natural images, for example, House, Peppers, Barbara, Monarch and C.man. And we compare the proposed GMMH algorithm with several state-of-the-art denoising methods, including NNM [28], BM3D [13], EPLL [17], LSSC [12], NCSR [29], SAIST [30], WNNM [31], BM3D<sub>bst</sub> that is the BM3D based on boosted patch searching method [20] denoising algorithms, SLR [23] and LR-GSC [32]. As a common experimental setting in the literature, we add white Gaussian noise with zero mean and standard deviation  $\sigma$  to the noise-free images and to test the performance of competing denoising methods. The MATLAB source code of our proposed algorithm can be downloaded at [https://github.com/KazukiAmakawa/HOSVD\\_denoise\\_patch](https://github.com/KazukiAmakawa/HOSVD_denoise_patch).

**Table 3** Parameter setting

$\sigma$	$\gamma$	$\beta$	$p$	$M$
10	0.28	0.16	7	100
30	0.3	0.1	7	100
50	0.35	0.1	9	70

**Table 4** Average PSNR scores by different methods

$\sigma$	10	30	50
NNM [28]	34.13	28.6	25.8
BM3D [13]	34.89	29.68	27.25
LSSC [12]	35.05	29.65	27.28
BM3D <sub>bst</sub> [20]	35.02	29.74	27.36
EPLL [17]	34.43	29.03	26.54
NCSR [29]	35.00	29.61	27.17
SAIST [30]	35.12	29.83	27.44
WNNM [31]	35.33	<b>30.05</b>	27.54
SLR [23]	34.93	29.64	27.42
LR-GSC [32]	<b>35.47</b>	30.01	27.61
Ours	35.09	29.92	<b>27.67</b>

Bold values indicate the best PSNR for all of the methods (the first column)

### 5.1 Implementation details

Our presented image denoising method contains two stages: the searching image similar patches stage and the denoising stage. In the denoising stage, there exist four parameters need to be set:  $\gamma$ ,  $\beta$ , the patch size  $p$ , the number  $M$  of patches in a tensor  $\mathcal{X}$ . These parameters pick different values when the standard variation  $\sigma$  of noise is different (shown in Table 3). Intuitively, if you want to exploit more global image information of image, number  $M$  of patches in a tensor should be set as big as possible.

### 5.2 Results and discussion

We evaluate the competing methods from peak signal to noise ratio (PSNR), structural similarity index measurement (SSIM), normalized cross-correlation coefficient (NCC) and visual quality viewpoints.

#### 5.2.1 PSNR

In Table 4, we present the average PSNR results on three noise levels  $\sigma = 10, 30, 50$  for image House, Peppers, Barbara, Monarch and C.man. Higher PSNR results on each noise level are highlighted in bold. From the results shown in Table 4, it can be seen that our proposed algorithm achieves much better PSNR results than NNM. Secondly, our proposed algorithm has higher PSNR values than BM3D, LSSC, EPLL and NCSR, BM3D<sub>bst</sub>, SLR and is only slightly inferior to WNNM and LR-GSC for low noise level.

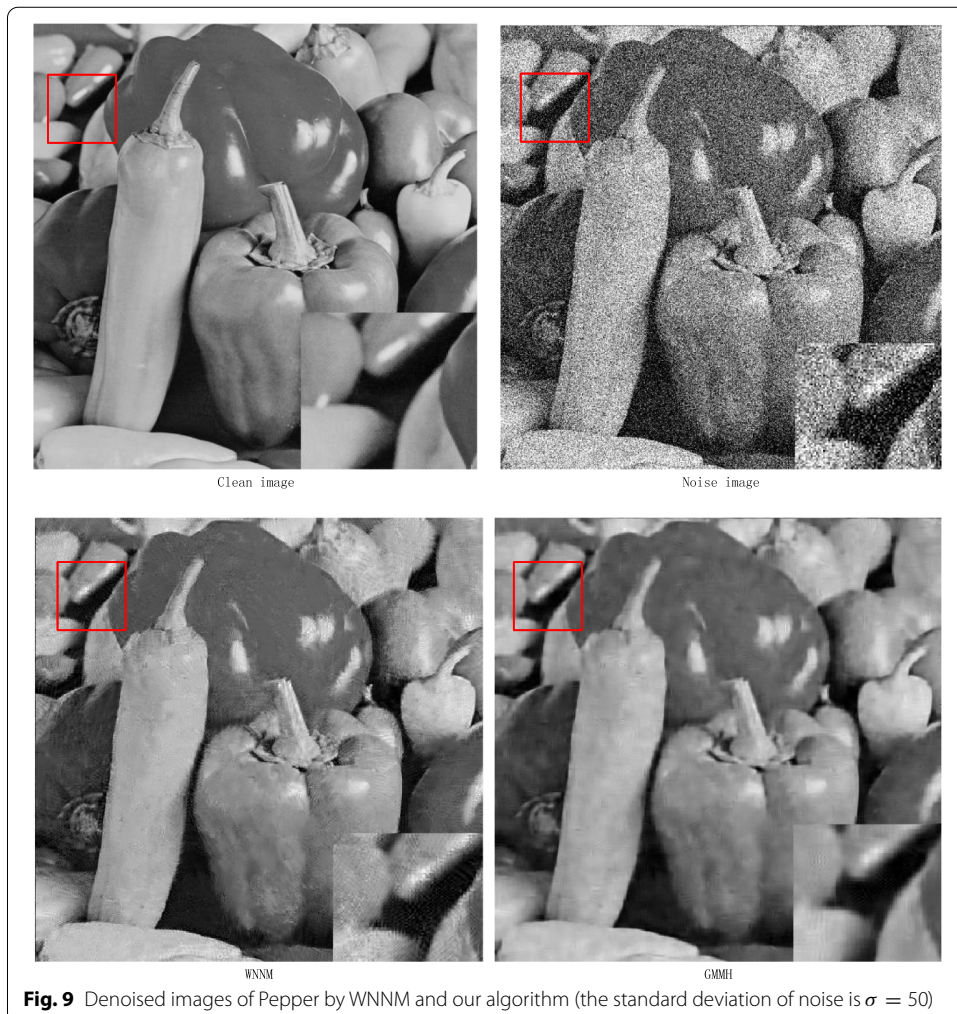


**Table 5** Average SSIM scores by different methods

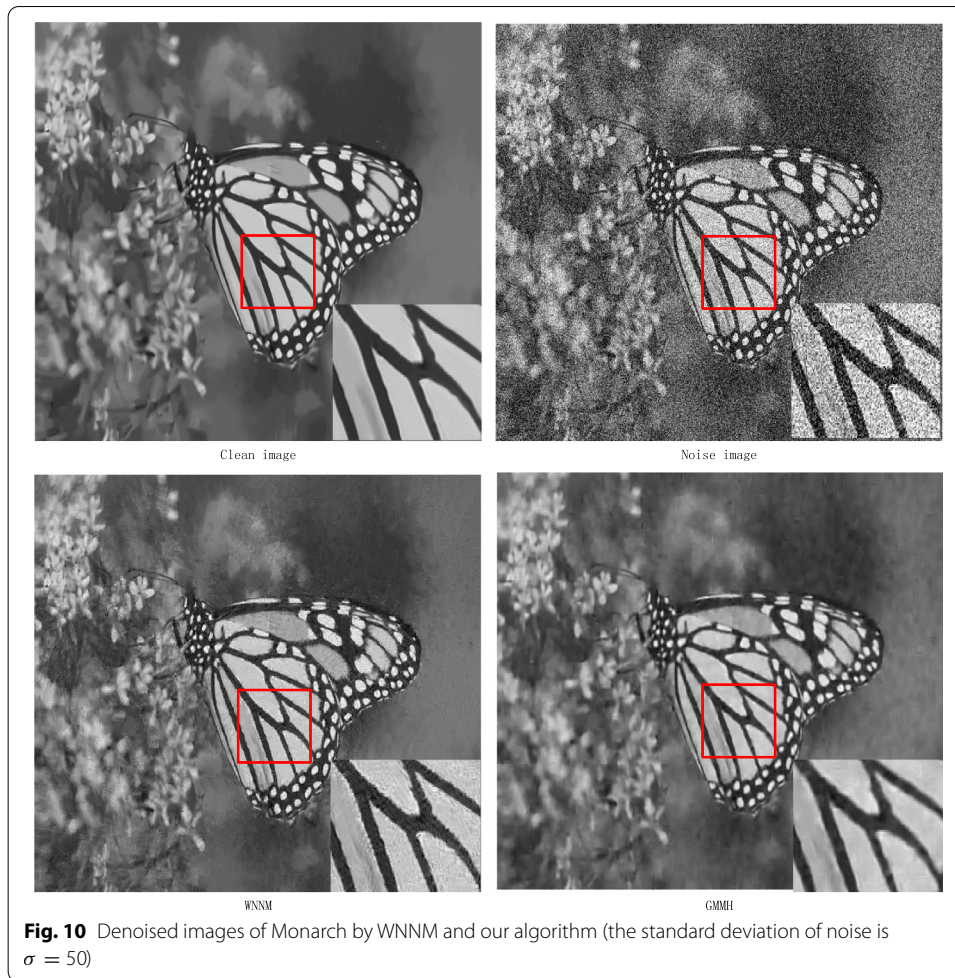
$\sigma$	20		40		60	
	SSIM	NCC	SSIM	NCC	SSIM	NCC
BM3D [13]	0.873	0.934	0.792	0.836	0.730	0.798
BM3D <sub>bst</sub> [20]	0.882	0.951	0.798	0.841	0.736	0.807
EPLL [17]	0.869	0.917	0.785	0.821	0.723	0.780
WNNM [31]	<b>0.884</b>	<b>0.952</b>	0.793	0.836	0.734	0.801
SLR [23]	0.878	0.938	0.792	0.836	0.731	0.798
LR-GSC [32]	0.883	0.951	<b>0.802</b>	<b>0.853</b>	0.734	0.801
Ours	0.881	0.951	0.797	0.841	<b>0.738</b>	<b>0.813</b>

Bold value indicates the best SSIM and NCC results for all of the methods (the first column)

With the noise level increasing, the performance of our proposed algorithm will be improved significantly, which can be verified for  $\sigma = 50$  noise level.



**Fig. 9** Denoised images of Pepper by WNNM and our algorithm (the standard deviation of noise is  $\sigma = 50$ )



### 5.2.2 SSIM & NCC

In Table 5, we report the average SSIM and NCC results on three noise levels  $\sigma = 20, 40, 60$  for image House, Peppers, Barbara, Monarch and C.man. We only compare our methods to BM3D, BM3D<sub>bst</sub>, EPLL, WNNM, SLR and LR-GSC, which outperforms the other algorithms. It can be seen from the SSIM viewpoint that our proposed method is comparable to these methods, and particularly, our method is superior to other method for  $\sigma = 50$  noise level. The NCCNCC (normalized cross-correlation) shows the similarity of the denoised image with original clear image. The range of NCC  $[-1, 1]$ , and the closer the 1 is, the more similar the two image are, the closer the  $-1$  indicates that the more dissimilar the two images. The NCC results in Table 5 show that there exists positive correlation between SSIM and NCC, which indicates that the higher SSIM value is, the bigger NCC value is.

### 5.2.3 Visual quality

Considering that human subjects are the ultimate judge of the image quality, the visual quality of denoised image is essential to evaluate a denoising algorithm. Figures 9 and 10 show the denoised images of Pepper and Monarch by the competing method. We obtain that WNNM has strong ability to image denoising by Table 4, so we just compare the

visual effect of WNNM with our proposed algorithm. It can be seen that the proposed algorithm in this paper is likely to generate artifacts than WNNM, but our algorithm preserves edge more better.

In summary, comparing to state-of-the-art denoising algorithm, our proposed method has performed a comparable job in image denoising, especially in high noise level.

## 6 Conclusion

In this paper, a new denoising approach based on image internal self-similarity prior, external patch priors and the low-rank patches tensor property of natural image is established. We search similar patches to each reference patch to form a tensor  $\mathcal{X}^{(k)}$ , and we decompose the tensor  $\mathcal{X}^{(k)}$  by HOSVD, then we use soft-threshold to tackle core tensor  $\mathcal{S}^{(k)}$ . In order to improve the performance of denoising, we have trained GMM by EM based on deep cluster. To enhance the accuracy for searching similar patches, a globally searching similar patches methods within the whole image is established and a geometric searching method is used to further improve the accuracy for finding similar patches.

### Abbreviations

HOSVD: High-order singular value decomposition; GMM: Gaussian mixture model; GPU: Graph process unit; BM3D: Block-matching and 3D filtering; WNNM: Weighted nuclear norm minimization; NNS: Nearest neighbor search; ED: Euclidean distance; EPLL: The expected patch log likelihood; EM: Expectation maximization algorithm; MD: Mahalanobis distance; MLE: Maximum likelihood estimation; BCE: Binary cross-entropy; SGD: The stochastic gradient descent; GKGLM: "GMM" + "k-means" + "inter-class geometry location similarity patches searching"; CP: CANDECOMP/PARAFAC decomposition; TNN: Tensor nuclear norm; SCAD: Smoothly clipped absolute deviation; MC: Minimax concave; GKGLM-Tensor: "GMM" + "k-means" + "inter-class geometry location similarity patches searching" + "Tensor".

### Acknowledgements

The work of the Zhengwei Shen was supported by Fundamental Research Funds for the Central Universities of China (FRF-GF-18-019sB).

### Authors' contributions

The first two authors have contributed equally to this manuscript!

### Funding

This manuscript was supported by Fundamental Research Funds for the Central Universities of China (FRF-GF-18-019B).

### Availability of data and materials

Not applicable.

## Declarations

### Competing interests

The authors declare that they have no competing interests.

### Author details

<sup>1</sup>School of Mathematics and Physics, University of Science and Technology Beijing, Beijing 100083, China. <sup>2</sup>North China Electric Power University, Beijing, China.

Received: 18 July 2019 Accepted: 14 September 2021

Published online: 31 December 2021

## References

1. G. Chang, B. Yu, M. Vetterli, Adaptive wavelet thresholding for image denoising and compression. *IEEE Trans. Image Process.* **9**(9), 1532–1546 (2000)
2. P. Chatterjee, P. Milanfar, Is denoising dead? *IEEE Trans. Image Process.* **19**(4), 895–911 (2010)
3. A. Levin, B. Nadler, Natural image denoising: optimality and inherent bounds, in *CVPR* (2011), pp. 2833–2840
4. A. Levin, B. Nadler, F. Durand, W.T. Freeman, Patch complexity, finite pixel correlations and optimal denoising, in *ECCV* (2012), pp. 73–86
5. C. Tomasi, R. Manduchi, Bilateral filtering for gray and color images, in *ICCV* (1998), pp. 839–846
6. K. Muhammad, T. Hussain, J. Del Ser, V. Palade, V.H.C. de Albuquerque, DeepReS: a deep learning-based video summarization strategy for resource-constrained industrial surveillance scenarios. *IEEE Trans. Ind. Inf.* **16**(9), 5938–5947 (2020)

7. A. Buades, B. Coll, J. Morel, A non-local algorithm for image denoising, in *CVPR* (2005), pp. 60–65
8. P. Chatterjee, P. Milanfar, Clustering-based denoising with locally learned dictionaries. *IEEE Trans. Image Process* **18**(7), 1438–1451 (2009)
9. P. Chatterjee, P. Milanfar, Patch-based near-optimal image denoising. *IEEE Trans. Image Process.* **21**(4), 1635–1649 (2012)
10. X. Chen, S.B. Kang, J. Yu, Fast patch-based denoising using approximated patch geodesic paths, in *CVPR* (2013), pp. 1211–1218
11. W. Dong, L. Zhang, G. Shi, X. Li, Nonlocally centralized sparse representation for image restoration. *IEEE Trans. Image Process.* **22**(4), 1620–1630 (2013)
12. J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Non-local sparse models for image restoration, in *ICCV* (2009), pp. 2272–2279
13. K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.* **16**(8), 2080–2095 (2007)
14. O. Lotan, M. Irani, Needle-match: reliable patch matching under high uncertainty, in *CVPR* (2016), pp. 439–448
15. V. Kumar BG, G. Carneiro, I. Reid, Learning local image descriptors with deep siamese and triplet convolutional networks by minimizing global loss functions, in *CVPR* (2016), pp. 5385–5394
16. S. Zagoruyko, N. Komodakis, Learning to compare image patches via convolutional neural networks, in *CVPR* (2015), pp. 4353–4361
17. D. Zoran, Y. Weiss, From learning models of natural image patches to whole image restoration, in *ICCV* (2011), pp. 479–486
18. C. Deledalley, S. Parameswaran, T.Q. Nguyen, Image denoising with generalized Gaussian mixture model patch priors. *SIAM J. Imaging Sci.* **11**(4), 2568–2609 (2018)
19. F. Chen, L. Zhang, H. Yu, External patch prior guided internal clustering for image denoising, in *ICCV* (2015), pp. 603–611
20. S. Lu, Good similar patches for image denoising, in *WACV* (2019), pp. 1886–1895
21. S. Gandy, B. Recht, I. Yamada, Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Probl.* **27**(2), 025010 (2011)
22. Z. Shen, H. Sun, Iterative adaptive nonconvex low-rank tensor approximation to image restoration based on ADMM. *J. Math. Imaging Vis.* **61**(5), 627–642 (2019)
23. D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in *ICCV*, vol. 2 (2001), pp. 416–423
24. L.D. Lathauwer, *Signal Processing Based on Multilinear Algebra* (Katholieke Universiteit Leuven, Leuven, 1997)
25. J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(1), 208–220 (2013)
26. Z. Zhang, G. Ely, S. Aeron, N. Hao, M. Kilmer, Novel methods for multilinear data completion and de-noising based on tensor-SVD, in *CVPR* (2014), pp. 3842–3849
27. K. Zhang, W. Zuo, S. Gu, L. Zhang, Learning deep CNN denoiser prior for image restoration, in *CVPR* (2017), pp. 2808–2817
28. M. Fazel, H. Hindi, S.P. Boyd, A rank minimization heuristic with application to minimum order system approximation, in *Proceedings of the 2001 American Control Conference*, vol. 6 (2001), pp. 4734–4739
29. W. Dong, L. Zhang, G. Shi, Centralized sparse representation for image restoration, in *ICCV* (2011), pp. 1259–1266
30. W. Dong, G. Shi, X. Li, Nonlocal image restoration with bilateral variance estimation: a low-rank approach. *IEEE Trans. Image Process.* **22**(2), 700–711 (2013)
31. S. Gu, L. Zhang, W. Zuo, X. Feng, Weighted nuclear norm minimization with application to image denoising, in *CVPR* (2014), pp. 2862–2869
32. Z. Zha, B. Wen, X. Yuan, J. Zhou, C. Zhu, Image restoration via reconciliation of group sparsity and low-rank models. *IEEE Trans. Image Process.* **30**, 5223–5238 (2021)
33. M.A.T. Figueiredo, A.K. Jain, Unsupervised learning of finite mixture models. *IEEE Trans. Image Process.* **24**(3), 381–396 (2002)
34. G. Celeux, G. Govaert, Gaussian parsimonious clustering models. *Pattern Recognit.* **28**(5), 781–793 (1995)
35. N. Friedman, S. Russell, Image segmentation in video sequences: a probabilistic approach, *CVPR* (1997), pp. 130–158
36. H. Permuter, J. Francos, I.H. Jermyn, Gaussian mixture models of texture and colour for image database retrieval, in *ICASSP* (2003), pp. III-569

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.