Multimedia Tools and Applications (2020) 79:33043–33069
https://doi.org/10.1007/s11042-020-09565-3

A mixed model with multi-fidelity terms and nonlocal low rank regularization for natural image noise removal

Yuepeng Li[1,2] · Chun Li[1,2]

Received: 1 January 2020 / Revised: 3 July 2020 / Accepted: 6 August 2020 /
Published online: 31 August 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Reconstructing an original image from its corrupted observation is an important and fundamental problem in many image processing applications. Generally, the L_1-norm or L_2-norm combined with a regularization term (the total variation (TV), total generalized variation (TGV) or nuclear norm) is used to fit the impulse noise and Gaussian noise, respectively. However, these methods can only be used to remove a single type of noise from images, and traditional regularization terms often have difficulties in capturing some important prior knowledge of images, such as nonlocal self-similarity, low rank and sparsity. To overcome the above issues, we propose a mixed noise removal model with L_1-L_2 fidelity terms and a popular nonlocal low-rank regularization term, which has been shown to have more effective image denoising performance than traditional regularization methods. To solve this model, the split Bregman iteration method (SBIM) is adopted to decompose the difficult minimization optimization problem into four simple subproblems. Extensive experiments on natural images demonstrate that the effectiveness of the proposed method is better than that of other state-of-the-art methods.

Keywords Mixed denoising model · Inverse problem · Alternative minimization · Image processing

1 Introduction

Reconstructing a true image from its poor-quality observation is one of the most important issues in image processing applications [5, 37, 38, 41]. Generally, we encounter two types of noise during the image acquisition process: impulse noise (IN) and additive white Gaussian noise (AWGN). IN is generally produced due to transmission errors and damaged pixels in camera sensors, and AWGN is generally produced during the image acquisition itself [4].

✉ Yuepeng Li
liyuepeng@cnic.cn

[1] Computer Network Information Center, Chinese Academy of Sciences, Beijing, 100190 China

[2] University of Chinese Academy of Sciences, Beijing, 100049 China

Many methods have been developed for removing either AWGN or IN. However, a mixture of IN and AWGN is normally encountered in real-life applications, which makes image denoising a challenging problem.

Formally, the aim of the denoising problem is to solve the inverse problem

$$Au = f \tag{1}$$

Here, A is an identity operator or blurring operator, and f is the known data (observed image) from the sensors, which may be of poor quality. $u : \Omega \subset R^d \to R$ denotes the unknown object (uncorrupted image).

Assume that u is an instance of a random variable U itself, and the corrupted image follows the conditional probability density $p(f; u) = p(f|U = u) = p(f|u)$. Given f, we can obtain the posteriori probability density for u based on the Bayesian formula $p(u|f) = \frac{p(f|u)p(u)}{p(f)}$.

To resolve u when it is polluted by Gaussian noise only, we can rewrite (1) as follows: $Au + \varepsilon = f, where \varepsilon \sim N(0, \sigma^2)$. Therefore, we can effectively remove the Gaussian noise by solving the optimization problem in (2), where we dropped every term that is independent of u.

$$\hat{u} = \underset{u \in L^1(\Omega)}{\arg\min} \left\{ \frac{1}{2\sigma^2} \left\| Au - f \right\|_{L^2(\Omega)}^2 - \log\left(p\left(u\right)\right) \right\} \tag{2}$$

Because Ω in (2) is bounded with a finite measure and $L^\infty(\Omega) \subset L^2(\Omega) \subset L^1(\Omega)$, the L_2-norm can fit the Gaussian noise [55]. In the last term $-\log(p(u))$, we can add some prior information about images u (e.g., piecewise smoothness [46] and the sparsity prior [14, 15, 35, 40, 41, 62, 64, 65]). We often choose the Gibbs function [25] as the prior, i.e., $p(u) \sim e^{-\lambda R(u)}$. Here, $R : L^1(\Omega) \to [0, \infty]$ is a convex energy function. Therefore, we can rewrite (2) as follows:

$$\hat{u} = \underset{u \in L^1(\Omega)}{\arg\min} \left\{ \left\| Au - f \right\|_{L^2(\Omega)}^2 + \lambda R\left(u\right) \right\} \tag{3}$$

Here, $R(u)$ is a regularization term.

For images polluted by IN only, which may be random-valued IN and salt-and pepper noise in real-life applications, we can replace the L_2 fidelity term $\|Au - f\|_{L^2(\Omega)}^2$ with L_1 fidelity, which performs well in removing outliers and IN. Then, we can obtain the following optimization problem for IN removal:

$$\hat{u} = \underset{u \in L^1(\Omega)}{\arg\min} \left\{ \left\| Au - f \right\|_{L^1(\Omega)} + \lambda R\left(u\right) \right\} \tag{4}$$

The aforementioned approaches mainly concentrate on AWGN or IN denoising. However, a mixture of AWGN and IN is frequently encountered in practice, which makes the problem of image noise reduction considerably more challenging due to the very different distributions of the two types of noises. Based on the above derivations for the image affected by either Gaussian noise or IN, we can define a flexible mixed noise removal framework as follows, and many previous mixed AWGN and IN noise removal methods [8, 26, 29, 52, 55] can be expressed under this framework:

$$\underset{u \in L^1(\Omega)}{\arg\min} \left\{ \begin{array}{l} \mu_1 \|Au - f\|_{L^1(\Omega)} \\ + \mu_2 \left\| Au - f \right\|_{L^2(\Omega)}^2 \\ + \lambda R\left(u\right) \end{array} \right\} \tag{5}$$

Here, μ_1, μ_2 and λ are weight parameters which balance the mixed noise and regularization term. The noise parameters are unknown and need to be estimated in real-world denoising applications. The L_1 and L_2 data fidelity terms in (5) restrain the solution of minimization problem (5) to preserve the origianl image information as much as possible by fitting the noise distribution. The L_1 data fidelity has been proved to be more appropriate for fitting the impulse noise, while L_2 data fidelity is offten used to modeling the Gaussian noise distribution. The regularization term $R(u)$ places limits on u and forces the solution of (5) to show the supposed properties of our prior knowledge.

Inspired by the different prior knowledge studied over the past decades, we proposed a variational model that combines various popular image denoising techniques. Specifically, L1 [57] and L2 [25] norms are used to capture the IN and the GN noise, respectively. The TV norm [46] is employed to preserve the edge information of the image with sharp discontinuity. In addition, to maintain the nonlocal similarity and sparsity of natural images, a nonlocal low-rank regularization (NLR) [17] term is introduced.

In the experiments, we compare our method with other state-of-the-art methods on natural images. The extensive experimental results have verified that the proposed method performs better than all compared methods.

The main contributions of this work are as follows: (1) A mixed noise removal model is proposed, which combines multifidelity terms, TV norm and NLR. (2) SBIM is used to solve the proposed model by decomposing the minimization problem of the model into multiple simple subproblems and obtains a solution to a large global optimization problem by coordinating the solutions of the subproblems. (3) The performance of visual quantitative indicators is improved compared with other state-of-the-art mixed noise removal methods.

The remainder of this paper is organized as follows. Section 2 describes some related works. The details of the proposed method are given in Section 3. Section 4 presents the convergence analysis of the proposed model. In Section 5, we present the experimental results that demonstrate our model's performance using a variety of real-world images. Finally, the conclusions are drawn in Section 6.

2 Related work

Numerous previous methods have been developed for image denoising over the past decades, and they can be roughly divided into three categories: filtering methods, variational optimization methods and machine learning methods.

(1). Filtering methods modify the noisy image such that the denoised image can maintain several characteristics in the spatial domain or dictionary domain. In the spatial domain, clean images normally maintain local similarity, nonlocal self-similarity (NSS) and low-rank properties. Local similarity refers to the fact that the pixel value is closely related to its surrounding neighbors, and the local filters estimate the original pixel of the noisy image by averaging the values of neighbors, such as a linear filter (Gaussian filter, mean filter) and a nonlinear filter (bilateral filter, median filter) [43]. NSS refers to the fact that for a given local patch in a natural image, one can find many patches that are similar to it across the image. To capture the NSS of images, nonlocal filters must first collect similar patches into groups and then remove noise according to the features of similar patches. For example, the nonlocal means filter (NLM) [5] takes the mean of all pixels in the image, weighted by how similar these pixels are to the target pixel. To remove AWGN and IN simultaneously, [54] introduced the robust outlyingness ratio (ROR) to NLM as an IN detection mechanism. [39] proposed a fast nonlocal means filter (FNLM), which accelerates the NLM algorithm by eliminating pixels that have small weights. [18] proposed an adaptive NLM that tunes the parameters of the NLM-based patch regularity. [21] proposed a parameter-free NLM that precomputes the optimal parameters of NLM in an image database.

Low rank assumes that the clean image matrix or patch matrices often have a low-rank property, and we can recover a low-rank matrix approximation (LRMA) as the denoising result from the noise observation. Low-rank matrix factorization (LRMF) and nuclear norm minimization (NNM) are two popular classes of methods for solving the LRMA problem. LRMF factorizes the noise image matrix into the product of two low-rank matrices while simultaneously constraining certain fidelity on the noisy image. For example, according to the Eckart–Young–Mirsky theorem [19], the basic LRMF problem under the Frobenius norm can be solved by truncating the rank of the singular value matrix to k. [47] proposed a weighted Frobenius norm to measure the deviation between the reconstructed image and noisy image and solved the new LRMA problem by an EM algorithm. Instead, the noisy image is decomposed into different parts, and NNM recovers the underlying low-rank matrix u from the corrupted image f by minimizing the nuclear norm of u. For example, [9] proved that many NNM problems can easily be solved in closed form by imposing a soft-thresholding operation on the singular values of the observation matrix. [30] proposed a weighted low-rank model (WLRM) for mixed noise removal, which assigned the residuals in the objective function different weights. [59] propose a truncated nuclear norm (TNNR) to approximate the rank function, which minimizes the k smallest singular values of the low-rank image matrix. [53] proposed a weighted Schatten p-norm minimization (WSNM) method that generalizes the NNM to the Schatten p-norm minimization with weights assigned to different singular values.

In the dictionary domain, noisy images or patches are projected into the dictionary domain, and the coding of clean images shows a sparse property. The methods mainly differ in how to choose the dictionary and how to sparsify the encoding. One approach to choose the dictionary is from a fixed set of transforms (steerable wavelet, curvelet, contourlets, bandlets). When the noisy image is transformed, the denoising method shrinks the coefficients of coding and reconstructs images through an inverse transform [3]. Another approach is learning the dictionary from an image database or patches. For example, K-SVD [41] learns the optimal dictionary by performing a k-means algorithm on the singular value decomposition of clean images, and once the dictionary is constructed, we can denoise the image by orthogonal matching pursuit (OMP). PCA filters construct orthogonal dictionaries by performing principal component analysis on image patches and then shrinks the coefficients of the principal components by different functions, such as PB-PCA, PGPCA, PLPCA and PHPCA [3]. [13] proposed a clustering-based method (K-LLD) that bridged dictionary-based approaches with regression-based frameworks.

The above methods mainly exploit single image characteristics to remove noise; however, some works are combine NSS, low rank and sparse coding. For example, BM3D [35] exploits NSS by grouping similar patches into a 3D block and collaborative filtering of the block in the transform domain. [40] proposed a method that extends and combines nonlocal means and sparse coding by learned simultaneous sparse coding (LSSC) such that similar patches share the same dictionary elements in their sparse decomposition. [51] proposed a powerful image model for mixed noise removal that connects low-rank methods with simultaneous sparse coding by spatially adaptive iterative singular-value thresholding (SAIST). The weighted encoding and sparse nonlocal regularization are unified in the weighted encoding model for mixed noise removal (WESNR) [29] by weighting the encoding residual to follow a Gaussian distribution. [27] grouped similar patches in the spatiotemporal domain and formulated the video restoration problem as a joint sparse and low-rank matrix approximation problem. [16] proposed nonlocally centralized sparse representation (NCSR), which exploits NSS to estimate the sparse coding coefficients of the original image.

(2) Variational optimization-based image denoising methods have been studied for several decades. Most of these works use different regularization terms to add prior knowledge on u. Traditional regularizers mainly include Tikhonov regularization [48], total variation (TV) regularization [11, 42, 46], total generalized variation (TGV) regularization [6, 34] and sparsity-based regularization [62, 64, 65]. For example, if we choose the regularization term in (3) as $R\left(u\right) = \|\nabla u\|_{L^1(\Omega)}$, then the resulting model will become the famous ROF model [46].

The TV-L1 model [57] improved ROF by replacing the squared L_2-norm in the data fidelity term with the robust L_1-norm. Although there is only a slight difference between the ROF and TV-L1, the TV-L1 model has the following advantages: 1) It is shown to be contrast invariant. 2) The TV-L1 model is considerably more effective at removing noise containing strong IN (e.g., salt-and-pepper noise). Because of the efficient performance of the TV-L1 model in removing strong IN, the L1-data fidelity term $\|f - Au\|_1$ has been widely used in related studies [31, 33, 45]. In addition, some scholars have improved the TV-L1 model by introducing a second regularization term, as in the following equation:

$$\hat{u} = \arg\min_u \left\{ \frac{\gamma_1}{2} \|f - Au\|_1 + \frac{\alpha}{2} \|\nabla u\left(x\right)\|_2^2 + \|\nabla u\left(x\right)\|_1 \right\} \tag{6}$$

Since (6) is nonconvex, it is difficult to directly solve the minimization problem. Cai et al. [10] proposed the following two-stage segmentation method based on the Mumford–Shah model which replaced the L1 norm with L2 norm.

$$\hat{u} = \arg\min_u \left\{ \frac{\gamma_1}{2} \|f - Au\|_2^2 + \frac{\alpha}{2} \|\nabla u\left(x\right)\|_2^2 + \|\nabla u\left(x\right)\|_1 \right\} \tag{7}$$

Here, γ_1 and α are positive weight parameters. To remove the mixed Gaussian noise and IN, a combined $L_1 - L_2$ data fidelity term was suggested, and Shi et al. [28] proposed the following model by combining (6) with (7):

$$\hat{u} = \arg\min_u \left\{ \begin{array}{c} \frac{\gamma_1}{2} \|f - Au\|_1 + \frac{\gamma_2}{2} \|f - Au\|_2^2 \\ + \frac{\alpha}{2} \|\nabla u\left(x\right)\|_2^2 + \|\nabla u\left(x\right)\|_1 \end{array} \right\} \tag{8}$$

Here, $\alpha > 0$, $\gamma_1 \geq 0$, and $\gamma_2 \geq 0$ are parameters for balancing the data fidelity terms and the regularization terms.

The sparse-based regularization models can be expressed as the following problem [26], where the regularization term enforces the code of result to be sparse in the dictionary domain. As described in the filtering method, the dictionary can be precomputed in an image database or built on corrupted patches.

$$\hat{u} = \arg\min_{u \in L^1(\Omega)} \left\{ \begin{array}{c} \|Au - f\|_{L^2(\Omega)}^2 \\ +\eta \sum_{i=1}^{M} \|R_i u - D\alpha_i\|_{L^2(\Omega)}^2 \\ +\lambda \sum_{i=1}^{M} \varphi\left(\alpha_i\right) \end{array} \right\} \tag{9}$$

Here, $R_i \in R^{n \times N}$ is the matrix that is extracted from an image patch with size $\sqrt{n} \times \sqrt{n}$ centered at the pixel position i, $D \in R^{n \times k}$ denotes the dictionary, and $\varphi\left(\alpha_i\right)$ denotes the sparsity penalty on the sparse codes $\alpha_i \in R^K$.

Since the sparse coding can be connected with a low-rank approximation [51], Dong et al. [17] proposed an NLR to maintain the low rank and sparsity of the image. Huang et al. [26] developed an efficient mixed noise removal model that exploits a Laplacian scale mixture (LSM) to model mixed noise and NLR to model prior knowledge. In the nonlocal rank regularization methods, the key is to solve the rank minimization subproblem, and it can be approximately solved by minimizing the following optimization problem [17]:

$$E\left(u,\varepsilon\right)=\log\det\left(u+\varepsilon I\right) \tag{10}$$

Here, $u\in R^{n\times n}$ is a symmetric positive semidefinite matrix, and ε denotes a small constant value.

(3) Machine learning methods have been applied to many domains over the past decade, and such methods can be exploited for image denoising in two different ways. The first approach employs the learned model to solve a subproblem of other image denoising methods more quickly and efficiently. For example, Zhang et al. [61] decomposed the AWGN denoising model into a fidelity-term-related subproblem and a denoising subproblem by half quadratic splitting (HQS) and then trained a set of CNN denoisers to solve the denoising subproblem. Wang et al. [50] decomposed the mixed denoising model into four subproblems by the alternating direction method of multipliers (ADMM), and then they used a CNN as a Gaussian denoiser for the u subproblem.

Another approach removes the noise directly based on the learned model. Such methods are typical supervised learning methods that train models in a clean-noise paired dataset, such as the traditional machine learning methods Markov random fields (MRFs) [36], linear regression [63] and multilayer perceptrons [7]. Recently, many deep learning methods have been proven to be powerful tools for image denoising. [60] proposed a blind Gaussian denoising convolutional neural network (DnCNN) that predicts the residual image between the noisy observation and the latent clean image. [2] proposed a block-matching convolutional neural network (BMCNN) method that contains the denoising and averaging operations on the image patches. [2] proposed a blind denoising method GAN-CNN that employs a generative adversarial network (GAN) to build paired training datasets for CNN [2]

3 Proposed mixed noise removal method

Based on (5), we can reformulate our proposed mixed noise removal model as follows:

$$\hat{u}=\underset{u\in L^1(\Omega)}{\arg\min}\left\{\begin{array}{l}\mu_1\|Au-f\|_{L^1(\Omega)}+\mu_2\|Au-f\|^2_{L^2(\Omega)}\\+\lambda_1\|\nabla u\|_{L^1(\Omega)}+\lambda_2\sum\limits_i rank\left(\tilde{R}_j u\right)\end{array}\right\} \tag{11}$$

Here, the regularization term $R(u)=(\|\nabla u\|_{L^1(\Omega)},\sum\limits_i rank\left(\tilde{R}_j u\right))$ combines the TV with NLR [17] and $\lambda=(\lambda_1,\lambda_2)$ denotes the weight parameter of TV and NLR regularization terms. $\tilde{R}_j u=\left[R_{j_0}u,R_{j_1}u,R_{j_2}u,...,R_{j_{m-1}}u\right]\in R^{n\times m}$ denotes the matrix formed by the similar patch set of the exemplar image patch $u_j\in R^n$ that is $\sqrt{n}\times\sqrt{n}$ in size and centered at position j, and $\tilde{R}_j$ is an operator to extract the similar patches.

The rank minimization problem can be approximated using the surrogate function $E\left(u,\varepsilon\right)$, which has been proven to be better than the nuclear norm [17]. For a general matrix

$X_i \in C^{m\times n}$, $n \leq m$, which is neither square nor positive semidefinite, Dong et al. [17] slightly modified (10) into the following:

$$\begin{aligned} L\left(X,\varepsilon\right) &= \log\det\left(\left(XX^T\right)^{\frac{1}{2}} + \varepsilon I\right) \\ &= \log\det\left((\Sigma)^{\frac{1}{2}} + \varepsilon I\right) \\ &= \sum_{r=1}^{r_0} \log\left(\sigma_r\left(X\right) + \varepsilon\right) \end{aligned} \tag{12}$$

Here, Σ is a diagonal matrix whose diagonal elements are eigenvalues of the matrix XX^T, i.e., $XX^T = U\Sigma U^{-1}$, $\Sigma^{1/2}$ is the diagonal matrix whose diagonal elements are the singular values of the matrix, and $\sigma_r\left(X\right)$ denotes the r^{th} singular value of X, $r_0 = \min\left(n, m\right)$. By combining the above energies, we formulate the nonlocal low-rank-based mixed noise removal model as follows:

$$\hat{u} = \underset{u\in L^1(\Omega)}{\arg\min}\left\{\begin{array}{c} \mu_1\|Au-f\|_{L^1(\Omega)} + \mu_2\|Au-f\|^2_{L^2(\Omega)} \\ +\lambda_1\|\nabla u\|_{L^1(\Omega)} + \lambda_2\sum_i L\left(\tilde{R}_j u,\varepsilon\right)\end{array}\right\} \tag{13}$$

Here, $L\left(\tilde{R}_j u, \varepsilon\right)$ refers to

$$\begin{aligned} L\left(\tilde{R}_j u,\varepsilon\right) &= \log\det\left(\left(\tilde{R}_j u\left(\tilde{R}_j u\right)^T\right)^{\frac{1}{2}} + \varepsilon I\right) \\ &= \log\det\left((\Sigma)^{\frac{1}{2}} + \varepsilon I\right) \\ &= \sum_{r=1}^{r_0} \log\left(\sigma_r\left(\tilde{R}_j u\right) + \varepsilon\right) \end{aligned} \tag{14}$$

Numerous methods can be used to solve model (13), such as the split Bregman iteration method (SBIM) [22], the primal-dual approach [12, 20], ADMM [23, 24] and the augmented Lagrange multipliers (ALM) method [32].

ALM transforms constrained optimization problems into unconstrained optimization problems by combining the Lagrange multiplier term with a quadratic penalty function. ADMM extends ALM by breaking the ALM optimization problem into several subproblems and obtaining the corresponding solutions alternately. However, in image processing, because the objective function generally contains L1 and L2 norms, the subproblems must optimize the L1 and L2 norms simultaneously, which is not an easy task. SBIM introduces several auxiliary variables and decouples the L1 and L2 norms into different subproblems. Therefore, the algorithm could solve the L1 subproblem with Bergen iteration and the L2 subproblem with other efficient convex optimization methods separately. In addition, SBIM uses less memory and is easy to code. Therefore, in this paper, we choose SBIM to solve (13).

We separate the optimization problem (13) into multiple simple subproblems. Specifically, we first introduce auxiliary variables $h = f - Au$, $d = \nabla u$ and $L_j = \tilde{R}_j u$, and then we turn (13) into the optimization problem (15).

$$\left\{\begin{array}{l} \arg\underset{u\in L^1(\Omega)}{\min}\left\{\begin{array}{l} \mu_1\|Au-f\|_{L^1(\Omega)} + \mu_2\|Au-f\|^2_{L^2(\Omega)} \\ +\lambda_1\|\nabla u\|_{L^1(\Omega)} + \lambda_2\sum_i L\left(\tilde{R}_j u,\varepsilon\right)\end{array}\right\} \\ \qquad\qquad s.t.\quad h = f - Au, d = \nabla u, L_j = \tilde{R}_j u \end{array}\right. \tag{15}$$

The augmented Lagrangian equation for (15) is as shown in (16):

$$\begin{aligned}
&L\left(u^{k+1}, h^{k+1}, d^{k+1}, L_j^{k+1}\right) \\
&= \mu_1 \|h\|_{L^1(\Omega)} + \mu_2 \|Au - f\|^2_{L^2(\Omega)} + \lambda_1 \|d\|_{L^1(\Omega)} \\
&\quad + \lambda_2 \sum_j L\left(L_j\right) + \gamma_1 \left\|h - (f - Au) - b_1^k\right\|^2_{L^2(\Omega)} \\
&\quad + \gamma_2 \left\|d - \nabla u - b_2^k\right\|^2_{L^2(\Omega)} + \gamma_3 \sum_j \left\|L_j - \tilde{R}_j u - U_j^k\right\|^2_{L^2(\Omega)}
\end{aligned} \tag{16}$$

Therefore, we solve the $u,\ h,\ d$, and L_j-subproblems as follows.

3.1 *u*-subproblem

By applying the variable splitting method to (16), u has the minimization problem (17).

$$u^{k+1} = \underset{u \in L^1(\Omega)}{\arg\min} \begin{pmatrix} \mu_2 \|Au - f\|^2_{L^2(\Omega)} \\ + \gamma_1 \left\|h - (f - Au) - b_1^k\right\|^2_{L^2(\Omega)} \\ + \gamma_2 \left\|d - \nabla u - b_2^k\right\|^2_{L^2(\Omega)} \\ + \gamma_3 \sum_j \left\|L_j - \tilde{R}_j u - U_j^k\right\|^2_{L^2(\Omega)} \end{pmatrix} \tag{17}$$

The optimal value of u^{k+1} must satisfy the following Euler-Lagrange equation:

$$\begin{aligned}
&(\mu_2 - \gamma_1)\, A^T A u - \gamma_2 \Delta u + \gamma_3 \sum_j \widetilde{R_j^T} \tilde{R}_j u \\
&= \mu_2 A^T f + \gamma_1 A^T \left(f - h + b_1^k\right) \\
&\quad - \gamma_2 div \left(d^k - b_2^k\right) + \gamma_3 \sum_j \widetilde{R_j^T} \left(L_j - U_j^k\right)
\end{aligned} \tag{18}$$

Here, $\widetilde{R_j^T} L_j = \sum_{r=0}^{m-1} R_{j_r}^T u_{j_r}$, $\widetilde{R_j^T} \tilde{R}_j = \sum_{r=0}^{m-1} R_{j_r}^T R_{j_r}$, and R_{j_r} is a matrix, which is the image patch in the j_r position. We can use the FFT technique or the Gauss-Seidel method to solve this equation. Therefore, (18) can be efficiently solved in the discrete Fourier domain by assuming a periodic boundary condition by denoting $\mathcal{F}(u)$ as the Fourier transform of u.

3.2 *h*-subproblem

Similar to the u-subproblem, h has the following minimization problem:

$$h^{k+1} = \underset{h \in L^1(\Omega)}{\arg\min} \left\{ \mu_1 \|h\|_{L^1(\Omega)} + \gamma_1 \left\|h - (f - Au) - b_1^k\right\|^2_{L^2(\Omega)} \right\} \tag{19}$$

This convex L_1-problem is quite easy to solve, and it has an approximate solution through the shrinkage operator [49]:

$$h^{k+1} = Shrink\left(f + b_1^k - Au^k, \frac{\mu_1}{\gamma_1}\right) \tag{20}$$

Here, $shrink\,(s, t)$ is the shrinkage operator, and it is defined at each point $\alpha \in [0, 1]^2$:

$$shrink\,(s_\alpha, t_\alpha) = \frac{s_\alpha}{|s_\alpha|} \max\left(|s_\alpha| - t_\alpha, 0\right) \tag{21}$$

3.3 d-subproblem

Because d is also a convex L_1-problem, the same procedure may easily be adapted to obtain

$$d^{k+1} = \arg \min_{d \in L^1(\Omega)} \left\{ \lambda_1 \|d\|_{L^1(\Omega)} + \gamma_2 \left\| d - \nabla u^{k+1} - b_2^k \right\|^2_{L^2(\Omega)} \right\} \tag{22}$$

As in (20), we can apply the shrinkage operator again and obtain the following closed-form solution:

$$d^{k+1} = Shrink\left(\nabla u^k + b_2^k, \frac{\lambda_1}{\gamma_2} \right) \tag{23}$$

3.4 L_j-subproblem

As a consequence of the variable splitting technique, L_j has the following minimization problem:

$$L_j^{k+1} = \arg \min_{L_j \in L^1(\Omega)} \left\{ \begin{array}{l} \lambda_2 \sum\limits_j L\left(\tilde{R}_j u, \varepsilon \right) \\ +\gamma_3 \sum\limits_j \left\| L_j - \tilde{R}_j u - U_j^k \right\|^2_{L^2(\Omega)} \end{array} \right\} \tag{24}$$

In summary, the optimization problem (24) is equivalent to the following minimization problem (25):

$$L_j^{k+1} = \arg\min_{L_j \in L^1(\Omega)} \left\{ \begin{array}{c} \lambda_2 \sum\limits_{r=1}^{r_0} \log\left(\sigma_r\left(L_j \right) + \varepsilon \right) \\ + \gamma_3 \left\| L_j - Y_j \right\|^2_{L^2(\Omega)} \end{array} \right\} \tag{25}$$

Here, $Y_j = \tilde{R}_j u$, $r_0 = \min(n, m)$, and $\sigma_r\left(L_j\right)$ denotes the r^{th} singular value of L_j. Let $g(\sigma) = \sum\limits_{j=1}^{n} \log\left(\sigma_j + \varepsilon\right)$. Then, $g(\sigma)$ can be formulated as follows:

$$g(\sigma) = g(\sigma^{(k)}) + \left\langle \nabla g(\sigma^{(\mathrm{k})}), \sigma - \sigma^{(k)} \right\rangle \tag{26}$$

Here, $\sigma^{(k)}$ is the solution in the k^{th} iteration. Therefore, we can rewrite (25) as follows:

$$L_j^{k+1} = \arg\min_{L_j \in L^1(\Omega)} \left\{ \lambda_2 \sum_{r=1}^{r_0} \frac{\sigma_j}{\sigma_j^{(k)} + \varepsilon} + \gamma_3 \left\| L_j - Y_j \right\|^2_{L^2(\Omega)} \right\} \tag{27}$$

For simplicity, we set up $\theta = \frac{\lambda_2}{\gamma_3}$, $\omega_j^{(k)} = \frac{1}{\left(\sigma_j^{(k)}+\varepsilon\right)}$, and $\phi\left(L_j,\omega\right) = \sum\limits_{j=1}^{r_0} \omega_j^{(k)}\sigma_j$. Consequently, we obtain the following formula:

$$L_j^{k+1} = \underset{L_j \in L^1(\Omega)}{\arg\min} \left\{ \left\| L_j - Y_j \right\|_{L^2(\Omega)}^2 + \theta\phi\left(L_j,\omega\right) \right\} \tag{28}$$

Theorem 1 (Proximal Operator of the Weighted Nuclear Norm) *For each $Y \in C^{m\times n}$, $0 \le \omega_1 \le \omega_2 \le ... \le \omega_{r_0}$ and $r_0 = \min(m,n)$, the minimization optimization problem is as follows:*

$$\min_{L\in L^1(\Omega)} \frac{1}{2} \left\| Y - L \right\|_{L^2(\Omega)}^2 + \theta\phi\left(L,\omega\right) \tag{29}$$

It can be changed to the following closed-form solution using the weighted singular value thresholding method:

$$S_{\omega,\theta}\left(Y\right) \overset{\Delta}{=} U(\Sigma - \theta diag\left(\omega\right))_+ V^T \tag{30}$$

Here, $U\Sigma V^T$ is the SVD of Y, and $(w)_+ \overset{\Delta}{=} \max(w,0)$. To obtain more details about the proof of Theorem 1, please reference the Appendix *of* [17].

Based on Theorem 1, the optimization problem (25) has a closed-form solution:

$$L_j^{k+1} = \mathrm{U}\left(\tilde{\Sigma} \text{ - } \frac{\lambda_2}{\gamma_3}\mathrm{diag}\left(\mathrm{w}^{(k)}\right)\right)_+ V^T \tag{31}$$

Here, $U\tilde{\Sigma}V^T$ is the SVD of $\tilde{R}_j u$, $\omega_j^{(k)} = \frac{1}{\left(\sigma_j^{(k)}+\varepsilon\right)}$, and $\sigma_r\left(L_j\right)$ denotes the r^{th} singular value of L_j.

3.5 Update multipliers

After resolving all subproblems, we also need to update the Lagrangian multipliers b_1, b_2, and U on the $(k+1)^{th}$ iteration using the following update equations:

$$\begin{aligned} b_1^{k+1} &= b_1^k + \left(f - Au^{k+1}\right) - h^{k+1} \\ b_2^{k+1} &= b_2^k + \nabla u^{k+1} - d^{k+1} \\ U_j^{k+1} &= U_j^k + \rho\left(L_j^{k+1} - \tilde{R}_j u^{k+1}\right) \end{aligned} \tag{32}$$

In this paper, we decompose the difficult optimization problem (15) into four subproblems (the u, h, d and L_j-subproblems) based on SBIM. All the subproblems have fast and accurate techniques for obtaining solutions. For example, the u-subproblem can be efficiently solved using the FFT technique or the Gauss-Seidel iteration method. The h-subproblem and d-subproblem can be successfully resolved by applying the shrinkage operator. Furthermore, the L_j-subproblem keeps the closed-form solution by using the weighted SVD. The optimization procedure is summarized in Algorithm 1.

Algorithm 1 Mixed noise removal algorithm with multifidelity terms and NLR.

```
initialization:
    (1) Set parameters $\lambda_1, \lambda_2, \mu_1, \mu_2, \gamma_1, \gamma_2$ and $\gamma_3$;
    (2) Initialize $u^0$ via median filtering, and set $h^0 = b_1^0 = 0, d^0 = b_2^0 = 0$;
    (3) Group similar patches for each $u_j$ using $u^0$;
for $t = 1, 2, 3, ..., T$ do
    for $k = 1, 2, 3, ..., K$ do
        (1) Solve problem (25) to obtain $L^k$;
        (2) Update $h$ via (20);
        (3) Update $d$ via (23);
        (4) Update $b_1, b_2$, and $U$ via (32);
        (5) Update $u$ via (18);
        (6) Output $u^{(t)} = u^{(k)}$, if $k = K$
    end
    (1) Group similar patches for each $u_j$ using $u^{(t)}$;
    (2) Update $\lambda_1, \lambda_2, \mu_1, \mu_2, \gamma_1, \gamma_2, \gamma_3$ with $\rho$ $(0 < \rho < 1)$;
    (3) Output the reconstruction image $\hat{u} = u^{(t)}$, if $t = T$;
end
```

4 Convergence analysis

In this section, we provide the convergence analysis of the proposed algorithm (Algorithm 1). The following convergence analysis is motivated by the theorem in [58]. We can view the unconstrained minimization problem (11) as the constrained problem (15).

Based on the operator splitting algorithm, the augmented Lagrangian function of the problem (15) is as shown in (33):

$$\begin{aligned}
&L\left(u^{k+1}, h^{k+1}, d^{k+1}, L_j^{k+1}, \eta, \eta_1, \eta_2\right)\\
&= \mu_1 \|h\|_{L^1(\Omega)} + \mu_2 \left\|Au - f\right\|_{L^2(\Omega)}^2\\
&\quad + \lambda_1 \|d\|_{L^1(\Omega)} + \lambda_2 \sum_j L\left(L_j\right) - \langle \eta, d - \nabla u\rangle\\
&\quad - \langle \eta_1, h - f + Au\rangle - \left\langle \eta_2, L_j - \tilde{R}_j u\right\rangle\\
&+ \gamma_1 \left\|h - (f - Au) - b_1^k\right\|_{L^2(\Omega)}^2\\
&+ \gamma_2 \left\|d - \nabla u - b_2^k\right\|_{L^2(\Omega)}^2\\
&+ \gamma_3 \sum_j \left\|L_j - \tilde{R}_j u - U_j^k\right\|_{L^2(\Omega)}^2
\end{aligned} \tag{33}$$

Here, η, η_1, and η_2 are dual variables. Set $x^* = \left(u^*, h^*, d^*, L_j^*, \eta^*, \eta_1^*, \eta_2^*\right)$ to be Karush-Kuhn-Tucker (KKT) conditions as follows:

$$\begin{cases} 0 \in \mu_1 \partial \|h^*\|_1 + \eta_1^* \\ 0 = \mu_2 A^T \left(Au^* - f\right) - div\eta_1^* \\ 0 \in \lambda_1 \partial \|d^*\|_1 + \eta^* \\ 0 \in \lambda_1 \partial \left(\sum\limits_j L\left(L_j\right)\right) + \eta_2^* \\ \int \eta_1^* dx = 0 \\ \int \eta_2^* dx = 0 \\ f - Au^* = h^* \\ \nabla u^* = d^* \\ L_j^* = \tilde{R}_j u^* \end{cases} \tag{34}$$

Theorem 2 *Set* $x^k = \left(u^k, h^k, d^k, L_j^k, b_1^k, b_2^k, U_j^k\right)$ *to be the iterates in Algorithm 1 and* $\tilde{x}^k = \left(u^k, h^k, d^k, L_j^k, \gamma_1 b_1^k, \gamma_2 b_2^k, \gamma_3 U_j^k\right)$. *Suppose that*

$$\lim_{k\to\infty} \left(x^k - x^{k-1}\right) = 0.$$

Then, any accumulation point of $\tilde{x}^k$ *is a KKT point of (34).*

Proof. First, the SBIM subproblem (16) of u is equivalent to (17), which leads to the optimality condition (18). From the optimality conditions for all variables, we can obtain equalities

$$\begin{cases} \left(\mu_2 - \gamma_1\right) A^T A \left(u^{k+1} - u^k\right) \\ -\gamma_2 \Delta \left(u^{k+1} - u^k\right) + \gamma_3 \sum\limits_j \tilde{R_j^T} \tilde{R}_j \left(u^{k+1} - u^k\right) \\ = \mu_2 A^T f + \gamma_1 A^T \left(f - h^k + b_1^k\right) \\ -\gamma_2 div \left(d^k - b_2^k\right) + \gamma_3 \sum\limits_j \tilde{R_j^T} \left(L_j^k - U_j^k\right) \\ -\left(\mu_2 - \gamma_1\right) A^T A u^k + \gamma_2 \Delta u^k - \gamma_3 \sum\limits_j \tilde{R_j^T} \tilde{R}_j u^k \\ h^{k+1} - h^k = Shrink\left(f + b_1^k - Au^k, \frac{\mu_1}{\gamma_1}\right) - h^k \\ d^{k+1} - d^k = Shrink\left(\nabla u^k + b_2^k, \frac{\mu_1}{\gamma_1}\right) - d^k \\ L_j^{k+1} - L_j^k = \mathrm{U}\left(\tilde{\Sigma} - \frac{\lambda_2}{\gamma_3} \mathrm{diag}\left(\mathrm{w}^{(k)}\right)\right)_+ V^T - L_j^k \\ b_1^{k+1} - b_1^k = \left(f - Au^{k+1}\right) - h^{k+1} \\ b_2^{k+1} - b_2^k = \nabla u^{k+1} - d^{k+1} \\ U_j^{k+1} - U_j^k = \rho\left(L_j^{k+1} - \tilde{R}_j u^{k+1}\right) \end{cases} \tag{35}$$

Because of $\lim\limits_{k\to\infty}\left(x^k-x^{k-1}\right)=0$, the right-hand side of each equality in (35) goes to zero as $k\to\infty$. Therefore, all the terms go to zero as $\to\infty$

$$\left\{\begin{array}{l}
(\mu_2-\gamma_1)\,A^TAu^k-\gamma_2\Delta u+\gamma_3\sum\limits_j\tilde{R_j^T}\tilde{R}_ju^k- \\
\begin{pmatrix}\mu_2A^Tf+\gamma_1A^T\left(f-h+b_1^k\right)\\ -\gamma_2div\left(d^k-b_2^k\right)+\gamma_3\sum\limits_j\tilde{R_j^T}\left(L_j^k-U_j^k\right)\end{pmatrix}\to 0 \\
\qquad Shrink\left(f+b_1^k-Au^k,\frac{\mu_1}{\gamma_1}\right)-h^k\to 0 \\
\qquad Shrink\left(\nabla u^k+b_2^k,\frac{\mu_1}{\gamma_1}\right)-d^k\to 0 \\
\qquad \mathrm{U}\left(\tilde{\Sigma}\text{ - }\frac{\lambda_2}{\gamma_3}\mathrm{diag}\left(\mathrm{w}^{(k)}\right)\right)_+V^T\text{ - }L_j^k\to 0 \\
\qquad \left(f-Au^{k+1}\right)-h^{k+1}\to 0 \\
\qquad \nabla u^{k+1}-d^{k+1}\to 0 \\
\qquad \left(L_j^{k+1}-\tilde{R}_ju^{k+1}\right)\to 0
\end{array}\right. \tag{36}$$

From (35), we can easily obtain the following equation:

$$\left\{\begin{array}{l}
(\mu_2-\gamma_1)\,A^TAu^*-\gamma_2\Delta u^*+\gamma_3\sum\limits_j\tilde{R_j^T}\tilde{R}_ju^* \\
=\mu_2A^Tf+\gamma_1A^T\left(f-h^*+b_1^*\right) \\
-\gamma_2div\left(d^*-b_2^*\right)+\gamma_3\sum\limits_j\tilde{R_j^T}\left(L_j^*-U_j^*\right) \\
\qquad h^*=Shrink\left(f+b_1^*-Au^*,\frac{\mu_1}{\gamma_1}\right) \\
\qquad d^*=Shrink\left(\nabla u^*+b_2^*,\frac{\mu_1}{\gamma_1}\right) \\
\qquad L_j^*=\mathrm{U}\left(\tilde{\Sigma}\text{ - }\frac{\lambda_2}{\gamma_3}\mathrm{diag}\left(\mathrm{w}^{(*)}\right)\right)_+V^T
\end{array}\right. \tag{37}$$

Therefore, any accumulation of $\tilde{x}^k$ satisfies the KKT conditions (34) due to (35). However, since the minimization problem (15) is nonconvex, the KKT conditions are the only necessary optimal conditions for (15). Therefore, we cannot ensure that an accumulation point of x^k is an optimal point of (33).

5 Experimental results

In this section, we evaluate the validity of the proposed model. We first introduce the process for selecting the model parameters, and then we compare the denoising effects with other models using different noise levels. We choose four image denoising methods as the baseline in our experiment: nonlocal filter method NLM [5], sparse coding method WESNR [29], nonlocal low regularization method LSM_NLR [26] and the recently proposed deep learning method BdCNN [1] for blind noise whose learned network model is open sourced on GitHub. Because NLM was originally proposed for AWGN, we only show the result of NLM in the AWGN noise removal experiment. We choose the following classic images in the experiments: **parrot, monarch, boat, cameraman, house and Lena** (Fig. 1).

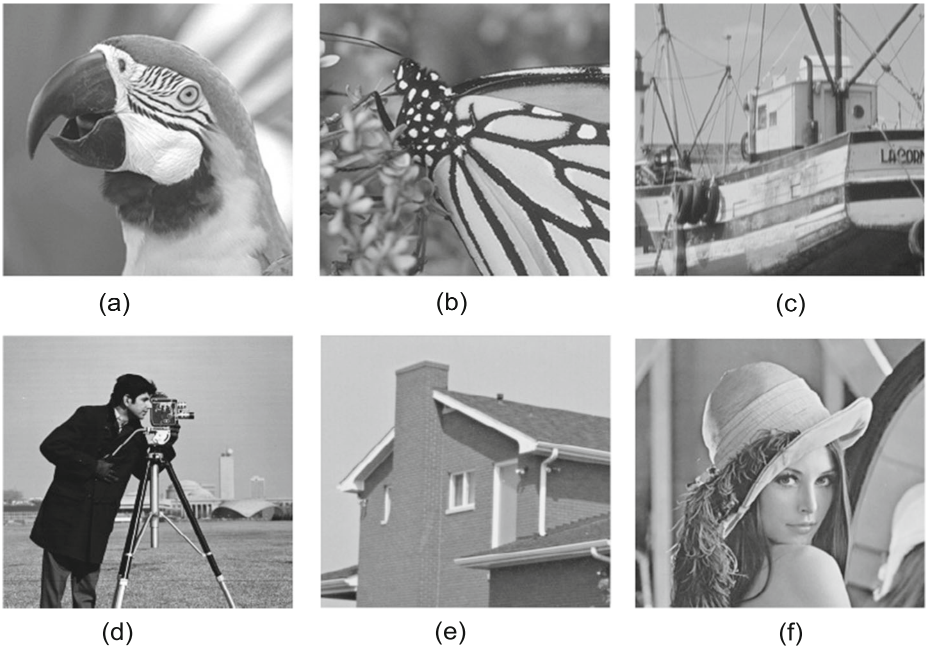

Fig. 1 The six classic ground truths (**a-f**): **a** is the parrot, **b** is the monarch, **c** is the boat, **d** is the cameraman, **e** is the house, **f** is Lena

We utilize the improvement in the signal-to-noise ratio (ISNR), the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) to evaluate the effectiveness of all methods for removing the noise, and the three metrics can be defined as follows:

$$ISNR\left(u_*, \hat{u}, \tilde{u}\right) = 10\log_{10}\left(\frac{\sum\left(u_* - \tilde{u}\right)^2}{\sum\left(u_* - \hat{u}\right)^2}\right)$$

$$PSNR\left(u_*, \hat{u}\right) = 10\log_{10}\left(\frac{255^2 mn}{\left\|u_* - \hat{u}\right\|_2^2}\right)$$

$$SSIM\left(u_*, \hat{u}\right) = \frac{2\mu_{u_*}\mu_u + C_1}{\mu_{u_*}^2\mu_u^2 + C_1} * \frac{2\sigma_{u_*}\sigma_u + C_2}{\sigma_{u_*}^2\sigma_u^2 + C_2} * \frac{2\sigma_{u_*}\sigma_u + C_3}{\sigma_{u_*}^2\sigma_u^2 + C_3}$$

where $u_* \in R^{m\times n}$ is the clean image, $\hat{u} \in R^{m\times n}$ is the restored image, $\tilde{u} \in R^{m\times n}$ is the contaminated image, μ_{u_*}, μ_u is the mean of u_*, u, σ_{u_*}, σ_u is the covariance and C_1, C_2, C_3 are constants. Note that a high value (ISNR, PSNR and SSIM) indicates a better restoration result.

5.1 Parameter selection

The proposed algorithm needs to choose seven parameters: $\mu_1, \mu_2, \lambda_1, \lambda_2, \gamma_1, \gamma_2$ and γ_3. Generally, there are two methods for parameter selection: 1) the first is that parameters are chosen based on the experience of the experimenters, and 2) the second method is to fix the values of some parameters and adjust other parameter values to obtain the optimal parameters [20, 22]. In this paper, we choose the second method to select our parameters. The appendix section shows a demo parameter selection process for GN0.01 noise removal, and in our real parameter selection process, we tried more values to select the optimal parameters. It can be found from the appendix that the parameter value of λ_2 affects the denoising result most, and therefore the parameter selection for other noise level can firstly fixed the value of λ_2 and then adjust the value of $\mu_1, \mu_2, \lambda_1, \gamma_1, \gamma_2$ respectively.

Because the proposed algorithm is not guaranteed to converge to the optimal value as the update operation continues to run, the loop iteration can be determined by the researcher's experience. For example, as other algorithms implement [26, 29], we can divide the parameter value into several regions and set different loop iteration times for each region. In addition, we can also collect experimental results on some image databases to determine the loop iteration. For example, we can first fix the inner loop iteration, record the PSNR value of every outer iteration step on 100 images and select the iteration times that make most images finally achieve the optimal performance. As shown in Fig. 2, the best loop iteration is 3 for the image database. Note that for other image databases, the best loop iteration may be different.

Both the proposed algorithm and the LSM_NLR algorithm have an NLR term and two layers of a loop structure in their algorithms. For fairness, we set the outer loop and inner loop iteration times of the two methods to be 3 and 4, respectively, in the successive experiments, and the main parameter of the nonlocal regularization term in the LSM_NLR model and the proposed model are the same at different noise levels. We also set the same patch size and number of patches as the LSM_NLR in [17]. We first select the optimal λ_2 value of the LSM_NLR and then iteratively determine the other parameters of the proposed model.

For NLM [5] and WESNR [29], the parameters are setted as in the published papers, which may lead to a diffrent result when the images in our experiments are not the same as in the original papers. The parameters of BdCNN [1] are trained on images database and we employed the network published by [1] in the successive experiments, which may slightly affects the denoising result due to the training image database. Table 1 shows the parameter values that are selected in this paper for different types and levels of noise.

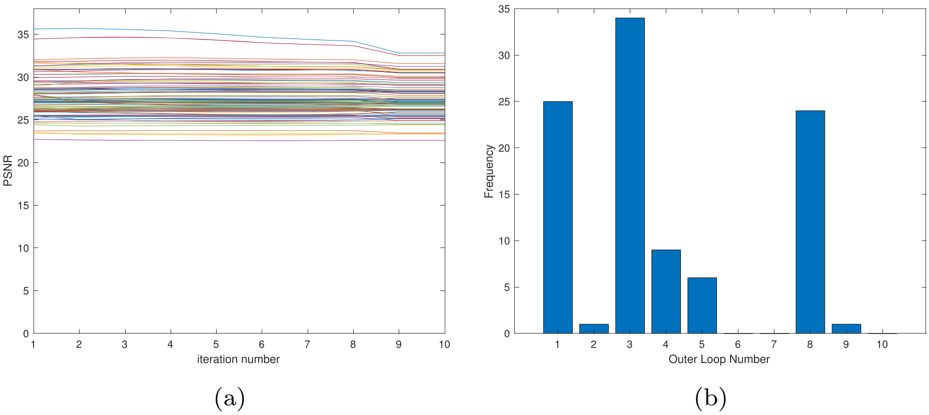

Fig. 2 **a** PSNRs of 10 iteration steps for 100 images **b** Frequencies of outer loop number which has best performace on given images

Table 1 The parameter selection

Noise / Parameter	λ_1	λ_2	μ_1	μ_2	γ_1	γ_2	γ_3
GN0.01	0.03	30	0.001	0.8	0.8	0.1	1.2
GN0.04	0.03	30	0.5	2.8	2.8	0.1	1.2
GN0.1	0.03	40	0.8	2.8	2.8	0.1	1.2
IN0.01	0.05	10	0.001	0.5	0.8	0.07	1.2
IN0.1	0.05	10	0.001	0.5	0.8	0.07	1.2
IN0.4	0.05	15	0.0005	0.008	0.008	0.07	1.2
GN0.01+IN0.01	0.03	30	0.001	0.8	0.8	0.1	1.2
GN0.01+IN0.1	0.03	30	0.000001	0.8	0.8	0.1	1.2
GN0.1+IN0.01	0.03	40	0.8	2.8	2.8	0.1	1.2

5.2 Experimental comparison of single noise

In this section, we compared the denoising effects of different models on images with a single type of noise. The types of noise are impulse noise(IN) and Gaussian noise(GN), which are added to the image using the MATLAB function *imnoise*, and the noise levels are (0.01, 0.04 and 0.1) and (0.01, 0.1 and 0.4), respectively.

The quantitative indicator (ISNR, PSNR and SSIM) results of the four models are shown in Tables 2 and 3. The GN0.04 denoising visualization effect of **parrots, monarch, boats, cameraman, house and Lena** is shown in Fig. 3. The IN0.4 *denoising* visualization effect of **parrots, monarch, boats, cameraman, house and Lena** is shown in Fig. 4.

As shown in Tables 2 and 3, for Gaussian noise, only the three GN0.01 polluted images' (**parrots, cameraman, Lena**) ISNR and PSNR values are lower than that of LSM_NLR, and the other 15 noise-added images are better than all the other models in the three indicators. For the IN noise, the denoising effect of the proposed model is significantly better than the other three models on all images, and the restored local visual quality is also better than the other models. Especially when IN noise level is low, the proposed method can not only preserve the nonlocal self-similarity (NSS) and nonlocal low-rank properties, but also distinguish and remove IN noises better than other methods. The BdCNN and WESNR methods remove all types of noise as same way by averaging the noise patches or restraining the low-rank property of denoising image, therefore the denoising result of these two methods are lower than that of LSM_NLR and the proposed method.

To investigate the algorithm, we count each step's PSNR value of the images in which our model's denoising effect is lower than the LSM_NLR (GN0.01 **parrot**), and we find that during the processing of these images, the PSNR value will first increase to the peak value in the first round of the inner loop and then follow an overall downward trend (see Fig. 5b). We plotted the PSNR values for each step of the two models for the **parrot** image with the GN0.01 noise level, and it can be observed that the PSNR values of both the LSM_NLR and our model (not just the proposed model) follow an overall downward trend during the iterative processing. However, for the situation in which the PSNR value of the proposed model is better than that of the LSM_NLR (e.g., **boat** image with GN0.01 noise), both models present an upward trend (see Fig. 5a). Therefore, we attempted several methods

Table 2 The quantitative indicators for the AWGN

<table>
<tr><th>Images</th><th>Methods</th><th>GN0.01 ISNR | PSNR | SSIM</th><th>GN0.04 ISNR|PSNR|SSIM</th><th>GN0.1 ISNR|PSNR| SSIM</th></tr>
<tr><td>parrots</td><td>OURS</td><td>9.21| 29.4 |**0.8739**</td><td>**6.9|26.59|0.8696**</td><td>**2.99|20.46|0.8321**</td></tr>
<tr><td></td><td>BdCNN [1]</td><td>8.23| 28.39 |0.7732</td><td>5.77| 25.48 |0.7565</td><td>2.42| 19.9 |0.7234</td></tr>
<tr><td></td><td>LSM_NLR [26]</td><td>**9.37**| **29.56** |0.8721</td><td>6.68| 26.37 |0.8676</td><td>2.83| 20.29 |0.8317</td></tr>
<tr><td></td><td>WESNR [29]</td><td>6.59| 26.78 |0.8334</td><td>5.28| 24.97 |0.829</td><td>2.33| 19.8 |0.8048</td></tr>
<tr><td></td><td>NLM [5]</td><td>7| 27.18 |0.6747</td><td>4.64| 24.29 |0.7918</td><td>2.87| 20.03 |0.7288</td></tr>
<tr><td>monarch</td><td>OURS</td><td>**9.06|29.09|0.9033**</td><td>**6.5|25.95|0.8967**</td><td>**2.76|19.91|0.8673**</td></tr>
<tr><td></td><td>BdCNN [1]</td><td>7.35| 27.38 |0.8071</td><td>5.2| 24.66 |0.8056</td><td>2.23| 19.36 |0.7634</td></tr>
<tr><td></td><td>LSM_NLR [26]</td><td>8.98| 29.01 |0.9014</td><td>6.25| 25.7 |0.8947</td><td>2.61| 19.76 |0.8668</td></tr>
<tr><td></td><td>WESNR [29]</td><td>5.25| 25.28 |0.8339</td><td>4.34| 23.79 |0.8246</td><td>2.06| 19.22 |0.807</td></tr>
<tr><td></td><td>NLM [5]</td><td>6.09| 26.17 |0.7501</td><td>3.54| 23.04 |0.7643</td><td>0.15| 19.56 |0.6031</td></tr>
<tr><td>boats</td><td>OURS</td><td>**9.69|29.75|0.8615**</td><td>**6.79|26.23|0.8582**</td><td>**2.81|19.89|0.8105**</td></tr>
<tr><td></td><td>BdCNN [1]</td><td>7.68| 27.72 |0.7661</td><td>5.65| 25.06 |0.7703</td><td>2.48| 19.53 |0.7548</td></tr>
<tr><td></td><td>LSM_NLR [26]</td><td>9.51| 29.57 |0.8603</td><td>6.44| 25.87 |0.8568</td><td>2.63| 19.71 |0.8109</td></tr>
<tr><td></td><td>WESNR [29]</td><td>7.68| 27.72 |0.7661</td><td>5.65| 25.06 |0.7703</td><td>2.48| 19.53 |0.7548</td></tr>
<tr><td></td><td>NLM [5]</td><td>6.29| 26.35 |0.7092</td><td>3.08| 22.48 |0.6339</td><td>0.65| 18.07 |0.4717</td></tr>
<tr><td>cameraman</td><td>OURS</td><td>6.9| 27.16 |**0.8107**</td><td>**5.33|24.84|0.8001**</td><td>**2.46|19.56|0.7392**</td></tr>
<tr><td></td><td>BdCNN [1]</td><td>6.41| 26.74 |0.6714</td><td>5| 24.52 |0.7324</td><td>2.33| 19.45 |0.7093</td></tr>
<tr><td></td><td>LSM_NLR [26]</td><td>**6.94**| **27.2** |0.8092</td><td>5.04| 24.56 |0.7992</td><td>2.28| 19.37 |0.7391</td></tr>
<tr><td></td><td>WESNR [29]</td><td>4.21| 24.47 |0.7558</td><td>3.2| 22.71 |0.734</td><td>1.44| 18.53 |0.6956</td></tr>
<tr><td></td><td>NLM [5]</td><td>6.25| 26.57 |0.6997</td><td>3.61| 23.12 |0.7172</td><td>0.81| 18.31 |0.61</td></tr>
<tr><td>house</td><td>OURS</td><td>**12.27|32.32|0.858**</td><td>**8.21|27.67|0.8575**</td><td>**3.27|20.41|0.834**</td></tr>
<tr><td></td><td>BdCNN [1]</td><td>9.55| 29.57 |0.7648</td><td>6.41| 25.81 |0.7614</td><td>2.73| 19.92 |0.6963</td></tr>
<tr><td></td><td>LSM_NLR [26]</td><td>12.14| 32.19 |0.8578</td><td>7.68| 27.14 |0.8574</td><td>3.02| 20.16 |0.8344</td></tr>
<tr><td></td><td>WESNR [29]</td><td>8.44| 28.49 |0.8128</td><td>5.32| 24.78 |0.8075</td><td>1.85| 18.99 |0.7939</td></tr>
<tr><td></td><td>NLM [5]</td><td>7.7| 27.75 |0.6541</td><td>4.83| 24.27 |0.7478</td><td>2.31| 18.75 |0.6355</td></tr>
<tr><td>Lena</td><td>OURS</td><td>**10.02|30.04|0.8764**</td><td>**7.12|26.54|0.8726**</td><td>**2.81|19.89|0.8285**</td></tr>
<tr><td></td><td>BdCNN [1]</td><td>8.02| 28.07 |0.7764</td><td>5.77| 25.18 |0.7678</td><td>2.48| 19.56 |0.7404</td></tr>
<tr><td></td><td>LSM_NLR [26]</td><td>10.19| 30.09 |0.8752</td><td>6.78| 26.2 |0.8708</td><td>2.65| 19.73 |0.829</td></tr>
<tr><td></td><td>WESNR [29]</td><td>6.21| 26.23 |0.784</td><td>4.52| 23.93 |0.779</td><td>1.76| 18.84 |0.7664</td></tr>
<tr><td></td><td>NLM [5]</td><td>6.49| 26.48 |0.6907</td><td>3.75| 23.15 |0.7101</td><td>1.37| 18.78 |0.573</td></tr>
</table>

The bold values denotes the indicators which perform best in the corresponding images

and finally found that adjusting the number of iterations can improve this situation. We set the number of inner loop iterations to 2, and the iterative PSNR values of the **parrot** with GN0.01 are shown in Fig. 5c. The PSNR values of both models show an overall upward trend, and the final PSNR value of our model is better than the LSM_NLR results.

In real applications, the real image and the PSNR value are unknown. The inner loop iteration number can be statistically determined by the best number in an image database, as shown in Fig. 3. Through this approach, although the selected loop iteration is not the best for every image, it can obtain the best performance for most noisy images in the same situation.

Table 3 The quantitative indicators for the in

Images	Methods	IN0.01 ISNR | PSNR | SSIM	IN0.04 ISNR | PSNR | SSIM	IN0.1 ISNR | PSNR | SSIM
parrots	OURS	**15.69|40.81|0.9984**	**23.31|38.47|0.994**	**24.49|33.7|0.9432**
	BdCNN [1]	13.64| 38.89 |0.9845	21.89| 37.03 |0.9729	23.77| 32.92 |0.9314
	LSM_NLR [26]	13.61| 38.74 |0.9495	22.65| 37.8 |0.9459	23.58| 32.8 |0.9044
	WESNR [29]	4.73| 29.86 |0.8945	14.73| 29.88 |0.8928	16.68| 25.9 |0.8437
monarch	OURS	**17.49|42.58|0.9993**	**23.15|38.45|0.9959**	**23.51|32.8|0.9665**
	BdCNN [1]	12.05| 37.63 |0.9896	21.13| 36.27 |0.9804	21.3| 30.5 |0.946
	LSM_NLR [26]	13.76| 38.86 |0.9761	21.85| 37.15 |0.9729	21.79| 31.08 |0.9381
	WESNR [29]	3.74| 28.84 |0.9216	13.27| 28.57 |0.9183	15.54| 24.83 |0.8343
boats	OURS	**16.3|42|0.9993**	**25.27|40.7|0.9969**	**25.41|34.79|0.9563**
	BdCNN [1]	14.34| 39.33 |0.9856	23.57| 38.89 |0.9802	24.07| 33.5 |0.9448
	LSM_NLR [26]	12.94| 38.63 |0.9641	22.48| 37.91 |0.9604	22.88| 32.27 |0.8995
	WESNR [29]	4.86| 30.56 |0.8803	14.83| 30.27 |0.8752	17.52| 26.9 |0.76
cameraman	OURS	**14.81|39.93|0.9983**	**20.58|35.62|0.9899**	**20.67|29.73|0.9272**
	BdCNN [1]	-3.03| 22.12 |0.7012	9.45| 24.5 |0.7138	18.45| 27.51 |0.8627
	LSM_NLR [26]	13.13| 38.26 |0.9537	20.84| 35.88 |0.9449	20.18| 29.23 |0.858
	WESNR [29]	1.84| 26.96 |0.8397	12.08| 27.12 |0.8373	14.52| 23.57 |0.7674
house	OURS	**15.88|41.42|0.9979**	**23.96|39.4|0.9938**	**26.93|36.4|0.9289**
	BdCNN [1]	15.1| 40.92 |0.9837	24.64| 40.13 |0.9746	25.75| 35.19 |0.9096
	LSM_NLR [26]	13.18| 38.73 |0.9373	22.92| 38.36 |0.9338	25.76| 35.24 |0.8942
	WESNR [29]	7.15| 32.7 |0.861	17.2| 32.63 |0.8608	20.34| 29.82 |0.819
Lena	OURS	16.62| 42.17 |**0.9988**	**24.73|40.21|0.9954**	**24.99|34.46|0.9537**
	BdCNN [1]	**13.6**| **38.98** |0.9878	23.13| 38.44 |0.98	23.62| 33.01 |0.9401
	LSM_NLR [26]	13.52| 39.08 |0.9642	22.59| 38.06 |0.9606	22.78| 32.25 |0.908
	WESNR [29]	4.65| 30.21 |0.8856	14.54| 30.01 |0.8815	17.65| 27.12 |0.8018

The bold values denotes the indicators which perform best in the corresponding images

5.3 Experimental comparison of mixed noise

In this subsection, we evaluate the effectiveness of the proposed model to remove the mixed noises. The mixed noises are IN and AWGN, and the noise levels are GN0.01+IN0.01, GN0.01+IN0.1 and GN0.1+IN0.01. The results for the denoising quantitative indicators (ISNR, PRSR and SSIM) of the four models are shown in Table 4, and the denoising visual effects of the four models with GN0.1+IN0.01 noise are shown for **parrots, monarch, boats, cameraman, house and Lena** in Fig. 6.

Similar to the denoising results of the Gaussian noise, it can be observed from Table 4 that all quantitative indicators of the proposed model algorithm in this paper is better than the other three models for the 16 images with mixed noise. In addition, the quality of the local visualization after recovery is better than that of other models. However, the ISNR and PSNR values of the GN0.01+IN0.01 and GN0.01+IN0.1 picture **parrots** are lower than that of the LSM_NLR model.

This situation mainly happens when Gaussian noise is low, and in such situation, the Laplacian scale mixture (LSM) part in LSM_NLR model is more suitable for modeling

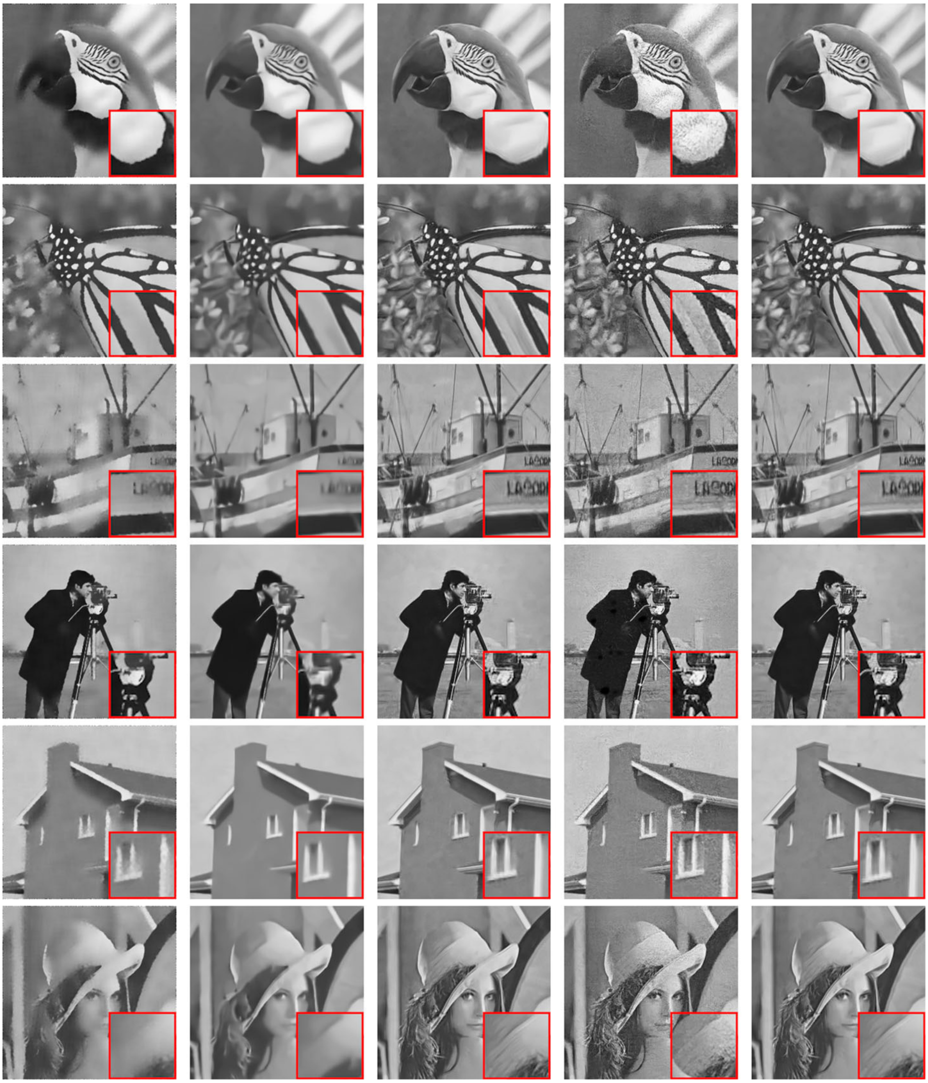

Fig. 3 Denoising effect of noisy images (GN0.1): The first column is the NLM, the second column is the WESNR, the third column is the LSM_NLR, the fourth column is BdCNN, and the fifth column is Oursx

Gaussian noise than the proposed model in some particular cases. We can validate that the denoising effect of GN0.01+IN0.01 and GN0.01+IN0.1 **parrots** can be improved by adjusting the number of iterations. Since WENSNR and BdCNN don't distinguish different types of noise in their noise removal process, the quantitative indicators of WENSNR and BdCNN are relatively lower for mixed noise image removal.

Although the denoising result of the proposed method is better than other models in most polluted images, the time efficiency of our model has no advantages. For example, the time consumed for GN0.01+IN0.01 **parrots** noise removal in our experimental computer

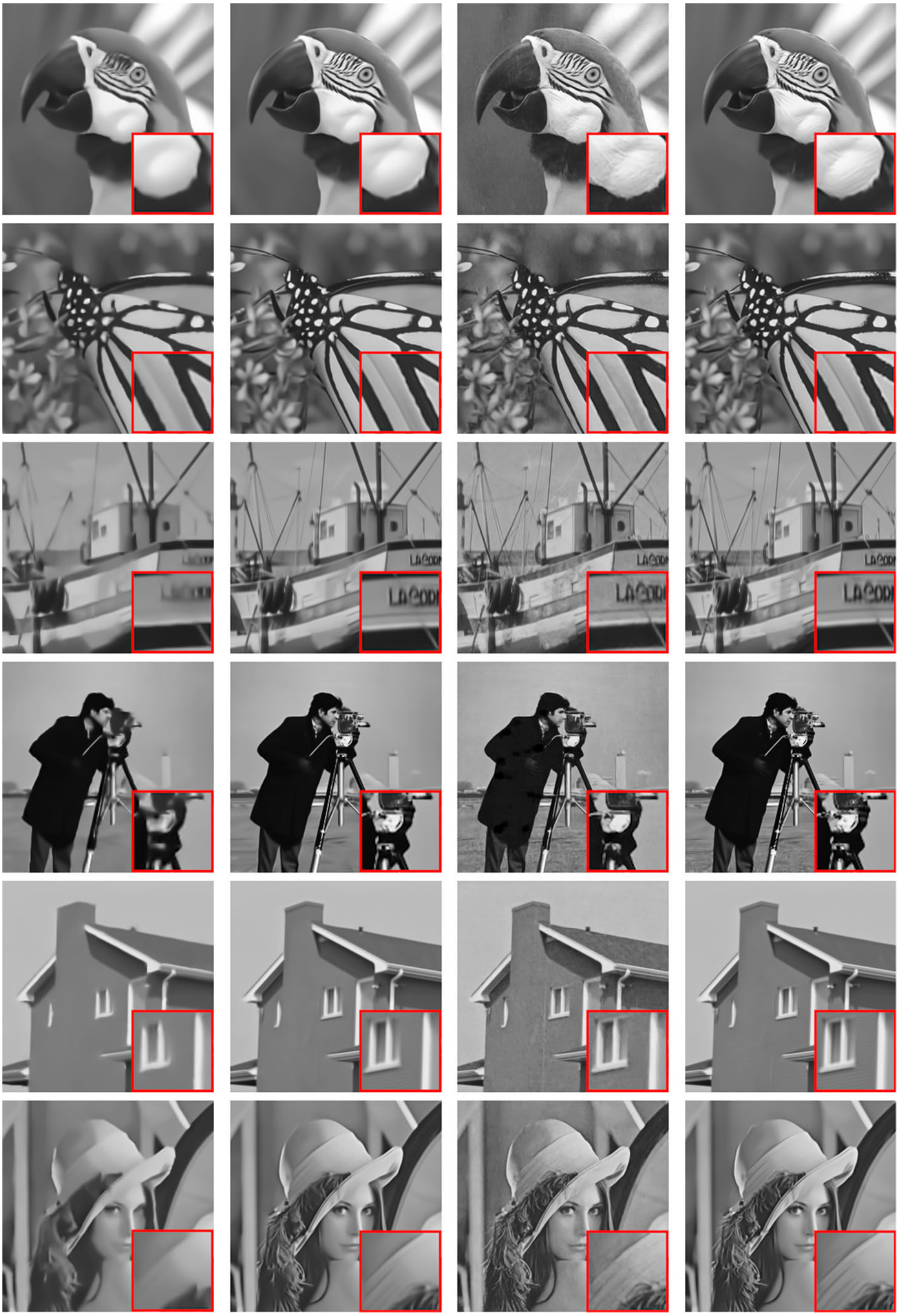

Fig. 4 Denoising effect of noisy images (IN0.4): The first column is the WESNR, the second column is the LSM_NLR, the third column is the BdCNN, and the fourth column is Ours

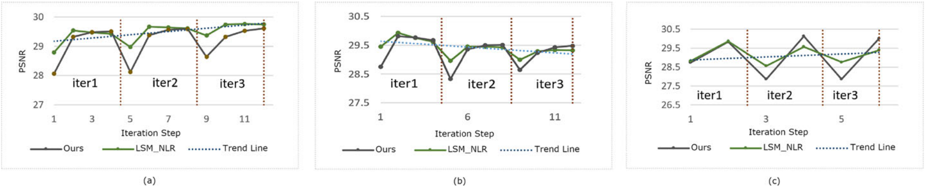

Fig. 5 PSNRs for different noisy images and iteration steps: **a** Boat with GN0.01 noise, **b** parrot with GN0.01 noise and **c** parrot with GN0.01 noise after adjusting the iteration step

is (OURS:54.6s,LSM_NLR:55.2s,WESNR:31.2s,BdCNN:15.7s). The running time of our algorithm is mainly consumed to solve the approximation low-rank optimization problem iterately, which has no advantage of time efficiency than other methods escpically for pre-trained method BdCNN. This shortcoming is also a limitation of the proposed method in

Table 4 The quantitative indicators for mixed noise

Images	Methods	GN0.01+IN0.01 ISNR | PSNR | SSIM	GN0.01+IN0.1 ISNR | PSNR | SSIM	GN0.1+IN0.01 ISNR | PSNR | SSIM
parrots	OURS	10.21| 29.23 |**0.8757**	15.12| 29.2 |**0.8727**	**3.57|20.42|0.8305**
	BdCNN [1]	9.42| 28.58 |0.794	14.69| 28.77 |0.8155	3.11| 19.92 |0.7524
	LSM_NLR [26]	**10.3**| **29.32** |0.8738	**15.23**| **29.32** |0.8719	3.4| 20.25 |0.83
	WESNR [29]	7.77| 26.8 |0.8335	12.54| 26.62 |0.8323	2.95| 19.79 |0.8056
monarch	OURS	**10.22|29.13|0.9049**	**14.49|28.67|0.9012**	**3.32|19.87|0.8655**
	BdCNN [1]	8.83| 27.77 |0.8328	13.29| 27.46 |0.8374	2.84| 19.38 |0.7834
	LSM_NLR [26]	10.13| 29.04 |0.9032	14.43| 28.61 |0.9004	3.17| 19.72 |0.865
	WESNR [29]	6.44| 25.36 |0.8326	11.06| 25.24 |0.8293	2.58| 19.13 |0.8038
boats	OURS	**10.81|29.73|0.8618**	**15.19|29.42|0.8548**	**3.35|19.89|0.8066**
	BdCNN [1]	9.17| 28.12 |0.7818	13.88| 28.08 |0.809	3.01| 19.51 |0.7643
	LSM_NLR [26]	10.63| 29.56 |0.8607	15.04| 29.27 |0.8547	3.17| 19.71 |0.8068
	WESNR [29]	7.31| 26.24 |0.7453	11.93| 26.16 |0.7433	2.5| 19.04 |0.7239
cameraman	OURS	**8.13**| 27.28 |**0.81**	**13.09**| 27.11 |**0.8024**	**3.02**| 19.48 |**0.7402**
	BdCNN [1]	7.4| 26.44 |0.6751	12.38| 26.44 |0.7454	2.94| 19.38 |0.7104
	LSM_NLR [26]	8.05| 27.19 |0.809	13| 27.02 |0.8027	2.84| 19.31 |0.7403
	WESNR [29]	5.34| 24.48 |0.756	10.16| 24.18 |0.7502	2.02| 18.49 |0.6953
house	OURS	**13.4**| 32.26 |**0.8577**	**17.81**| 32.13 |**0.8548**	**3.73**| 20.38 |**0.8356**
	BdCNN [1]	11.03| 29.91 |0.7772	15.96| 30.27 |0.8067	3.26| 19.88 |0.7219
	LSM_NLR [26]	13.16| 32.02 |0.8577	17.7| 32.02 |0.8551	3.5| 20.14 |0.8357
	WESNR [29]	9.55| 28.4 |0.8125	13.86| 28.18 |0.8101	2.34| 18.99 |0.793
Lena	OURS	**11.17**| 30.1 |**0.8786**	**15.56**| 29.88 |**0.8731**	**3.42**| 19.95 |**0.8279**
	BdCNN [1]	9.54| 28.52 |0.7975	14.2| 28.47 |0.8191	2.98| 19.58 |0.7535
	LSM_NLR [26]	11.02| 29.95 |0.8771	15.43| 29.75 |0.873	3.26| 19.78 |0.8282
	WESNR [29]	7.35| 26.28 |0.7866	11.94| 26.26 |0.7863	2.33| 18.85 |0.7643

The bold values denotes the indicators which perform best in the corresponding images

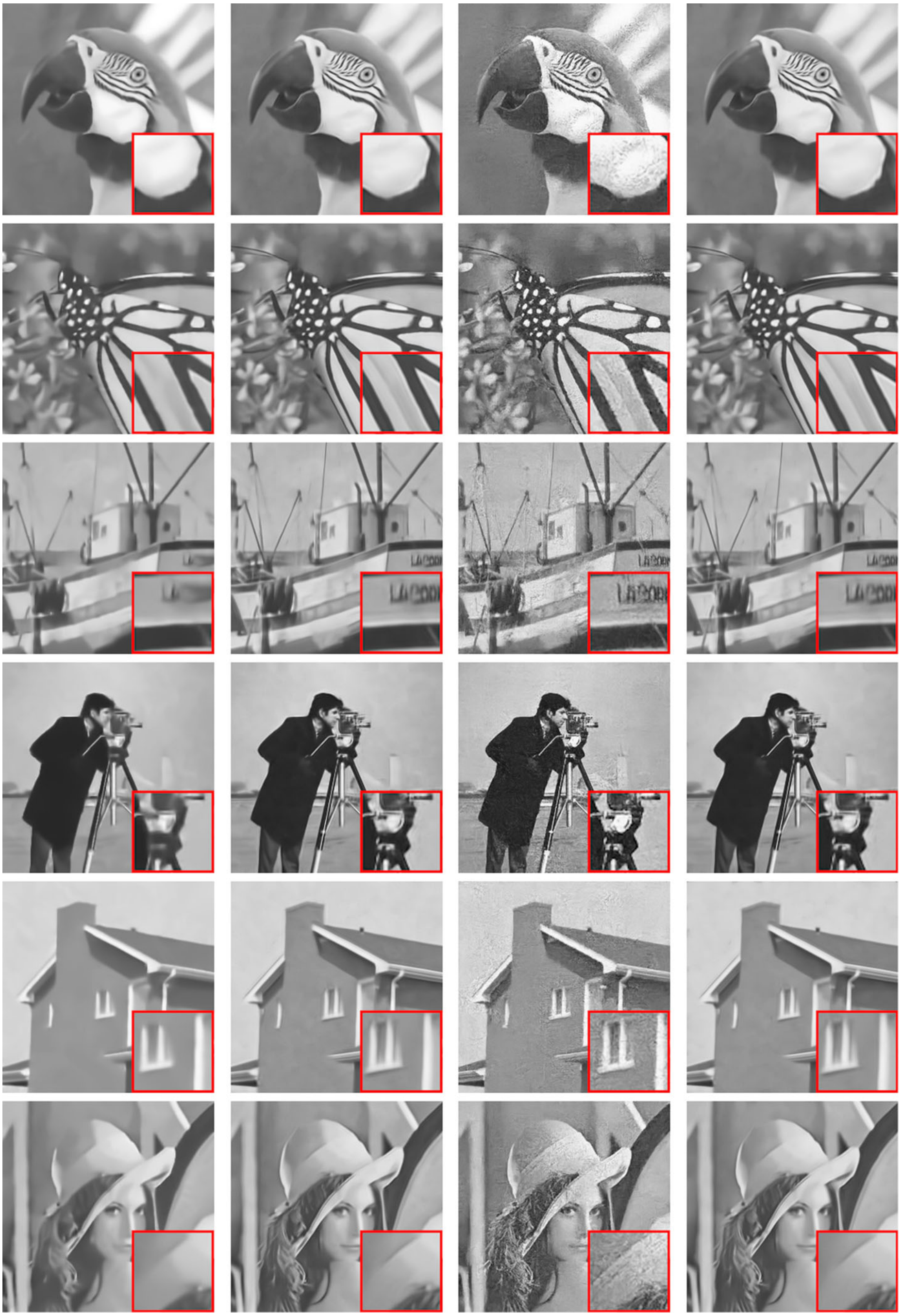

Fig. 6 Denoising effect of noisy images (GN0.1+IN0.01): The first column is the WESNR, the second column is the LSM_NLR, the third column is the BdCNN, and the fourth column is Ours

real applications, which could be improved by combining the proposed method with other pretrained methods or more efficient optimization methods.

6 Conclusion

In this paper, we proposed a mixed noise removal model that combines multifidelity terms with the nonlocal low-rank and TV regularization terms. In the proposed model, the L1 and L2 fidelity terms are employed to fit the IN and Gaussian noise, respectively. TV and NLR regularization impose edge, NSS, low-rank and sparsity prior knowledge on the natural images. We also use the nonconvex function logdet(X) as a smooth surrogate function for the rank approximation rather than the convex nuclear norm. To solve this model, we use SBIM to decompose the difficult minimization problem of (15) into four simple subproblems. Through many experiments on many images and different noise levels, it is shown that the ISNR, PSNR and SSIM values on almost noisy images of the proposed model are better than the values of four state-of-the-art models. In addition, the proposed model can retain the local information of the images and provide a better visualization effect.

However, there are still several limitations to our current study. Firstly, in some cases, the PSNR indicator follows an overall downward trend as the iterations increase. This problem can be solved to some extent by adjusting the number of iterations. However, determining this situation during the runtime processing of the algorithm and automatically adjusting the number of iterations is an important issue to be solved in our future work. Secondly, the time efficiency need to be further improved for real application, and we can utilize Laplacian smoothing-gradient descent [44] or other methods to speed up the objective function optimization process. Thirdly, since this study follows a traditional image processing approach, the parameters can not be setted automatically to adapt to various noise level. Therefore, in the future, we could exploit the deep split Bregman algorithm such as ADMM-Nets [56] to solve this problem, which combines the deep learning methods with traditional variational optimization model.

Appendix: Demo parameters selection process for GN0.01 noise removal

Steps	Parameters ($\lambda_1, \lambda_2, \mu_1, \mu_2, \gamma_1, \gamma_2, \gamma_3$)	Images	ISNR	PSNR	SSIM
Start point by experience	(0.01,20,0.01,0.01, 0.1,0.1,1)	parrots	6.02	26.23	0.6169
		monarch	5	25.04	0.6589
		boats	5.27	25.3	0.6481
		cameraman	4.78	25.03	0.616
		house	6.75	26.78	0.582
		Lena	5.59	25.62	0.6159
adjust γ_3	(0.01,20,0.01,0.01,0.1,0.1,**1.2**)	parrots	6.04	26.26	0.622
		monarch	5.06	25.12	0.6635
		boats	5.21	25.22	0.6464
		cameraman	4.82	25.12	0.63
		house	6.78	26.8	0.5858
		Lena	5.54	25.54	0.6115

adjust γ_2	(0.01,20,0.01,0.01,0.1,**0.2**,1.2)	parrots	5.93	26.14	0.6049
		monarch	5.05	25.1	0.6652
		boats	5.29	25.34	0.6486
		cameraman	4.72	24.97	0.6126
		house	6.7	26.71	0.5773
		Lena	5.51	25.51	0.6083
adjust γ_1	(0.01,20,0.01, 0.01,**0.8**,0.1,1.2)	parrots	6.06	26.29	0.6162
		monarch	5.02	25.08	0.6667
		boats	5.28	25.34	0.6495
		cameraman	4.79	25.07	0.6248
		house	6.78	26.82	0.5863
		Lena	5.55	25.56	0.6121
adjust μ_2	(0.01,20,0.01, **0.8**,0.8,0.1,1.2)	parrots	6.16	26.38	0.6306
		monarch	5.08	25.14	0.6697
		boats	5.33	25.42	0.6567
		cameraman	4.78	25.04	0.6229
		house	6.73	26.73	0.5781
		Lena	5.5	25.47	0.6063
adjust μ_1	(0.01,20,**0.1**,0.8,0. 8, 0.1,1.2)	parrots	5.99	26.18	0.6124
		monarch	4.99	25.01	0.659
		boats	5.21	25.23	0.6437
		cameraman	4.8	25.08	0.6279
		house	6.76	26.78	0.5843
		Lena	5.62	25.65	0.6213
adjust λ_2	(0.01,**25**,0.1,0.8,0.8, 0.1,1.2)	parrots	7.24	27.44	0.7087
		monarch	6.44	26.54	0.7389
		boats	6.6	26.66	0.714
		cameraman	5.69	25.93	0.6764
		house	8.49	28.49	0.6731
		Lena	7.13	27.19	0.7043
adjust λ_2	(0.01,**30**,0.1,0.8,0.8,0.1, 1.2)	parrots	9.1	29.3	0.8614
		monarch	8.81	28.86	0.885
		boats	9.41	29.43	0.8478
		cameraman	6.88	27.16	0.8025
		house	11.68	31.65	0.8405
		Lena	9.77	29.8	0.8615
adjust λ_1	(**0.03**,30,0.1,0.8,0.8,0.1,1.2)	parrots	9.27	29.45	0.8606
		monarch	8.76	28.84	0.8854
		boats	9.24	29.31	0.8501
		cameraman	6.87	27.18	0.8075
		house	11.76	31.75	0.8418
		Lena	9.81	29.89	0.8692

References

1. Abiko R, Ikehara M (2019) Blind denoising of mixed gaussian-impulse noise by single cnn. In: ICASSP 2019-2019 IEEE International conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 1717–1721
2. Ahn B, Cho NI (2017) Block-matching convolutional neural network for image denoising, arXiv:1704.00524
3. Alkinani MH, El-Sakka MR (2017) Patch-based models and algorithms for image denoising: a comparative review between patch-based images denoising methods for additive noise reduction. EURASIP J Image Video Process 2017(1):1–27
4. Bovik AC (2010) Handbook of image and video processing. Academic press
5. Buades A, Coll B, Morel JM (2005) A non-local algorithm for image denoising. In: IEEE Computer society conference on computer vision and pattern recognition
6. Bredies K, Kunisch K, Pock T (2010) Total generalized variation. SIAM J Imaging Sci 3(3):492–526
7. Burger HC, Schuler CJ, Harmeling S (2012) Image denoising with multi-layer perceptrons, part 1: comparison with existing algorithms and with bounds, arXiv:1211.1544
8. Cai J-F, Chan R, Nikolova M (2008) Two-phase approach for deblurring images corrupted by impulse plus gaussian noise. Inverse Probl Imaging 2(2):187–204
9. Cai J-F, Candès EJ, Shen Z (2010) A singular value thresholding algorithm for matrix completion. SIAM J Optim 20(4):1956–1982
10. Cai X, Chan R, Zeng T (2013) A two-stage image segmentation method using a convex variant of the mumford–shah model and thresholding. SIAM J Imaging Sci 6(1):368–390
11. Chambolle A (2004) An algorithm for total variation minimization and applications. J Math Imaging Vis 20(1-2):89–97
12. Chambolle A, Pock T (2011) A first-order primal-dual algorithm for convex problems with applications to imaging. J Math Imaging Vis 40(1):120–145
13. Chatterjee P, Milanfar P (2009) Clustering-based denoising with locally learned dictionaries. IEEE Trans Image Process 18(7):1438–1451
14. Deng L-J, Feng M, Tai X-C (2019) The fusion of panchromatic and multispectral remote sensing images via tensor-based sparse modeling and hyper-laplacian prior. Inf Fusion 52:76–89
15. Dong W, Xin L, Lei Z, Shi G (2011) Sparsity-based image denoising via dictionary learning and structural clustering. In: Computer vision & pattern recognition
16. Dong W, Zhang L, Shi G, Li X (2013) Nonlocally centralized sparse representation for image restoration. IEEE Trans Image Process 22(4):1620–1630
17. Dong W, Shi G, Li X, Ma Y, Huang F (2014) Compressive sensing via nonlocal low-rank regularization. IEEE Trans Image Process 23(8):3618–3632
18. Duval V, Aujol J-F, Gousseau (2010) On the parameter choice for the non-local means
19. Eckart C, Young G (1936) The approximation of one matrix by another of lower rank. Psychometrika 1(3):211–218
20. Ferstl D, Reinbacher C, Ranftl R, Rüther M, Bischof H (2013) Image guided depth upsampling using anisotropic total generalized variation. In: Proceedings of the IEEE International Conference on Computer Vision, pp 993–1000
21. Froment J (2014) Parameter-free fast pixelwise non-local means denoising. Image Process Line 4:300–326
22. Goldstein T, Osher S (2009) The split bregman method for l1-regularized problems. SIAM J Imaging Sci 2(2):323–343
23. He B, Yang H, Wang S (2000) Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. J Optim Theory Appl 106(2):337–356
24. He B, Tao M, Yuan X (2012) Alternating direction method with gaussian back substitution for separable convex programming. SIAM J Optim 22(2):313–340
25. Hebert T, Leahy R (1989) A generalized em algorithm for 3-d bayesian reconstruction from poisson data using gibbs priors. IEEE Trans Med Imaging 8(2):194–202
26. Huang T, Dong W, Xie X, Shi G, Bai X (2017) Mixed noise removal via laplacian scale mixture modeling and nonlocal low-rank approximation. IEEE Trans Image Process 26(7):3171–3186
27. Ji H, Huang S, Shen Z, Xu Y (2011) Robust video restoration by joint sparse and low rank matrix approximation. SIAM J Imaging Sci 4(4):1122–1142
28. Jia T, Shi Y, Zhu Y, Wang L (2016) An image restoration model combining mixed l1/l2 fidelity terms. J Vis Commun Image Represent 38:461–473

29. Jiang J, Zhang L, Yang J (2014) Mixed noise removal by weighted encoding with sparse nonlocal regularization. IEEE Trans Image Process 23(6):2651–2662
30. Jiang J, Yang J, Cui Y, Luo L (2015) Mixed noise removal by weighted low rank model. Neurocomputing 151:817–826
31. Jung M (2015) Kang Simultaneous cartoon and texture image restoration with higher-order regularization. SIAM J Imaging Sci 8(1):721–756
32. Jung M (2017) Piecewise-smooth image segmentation models with [formula] data-fidelity terms. J Sci Comput 70(3):1229–1261
33. Jung M, Kang M (2015) Efficient nonsmooth nonconvex optimization for image restoration and segmentation. J Sci Comput 62(2):336–370
34. Knoll F, Bredies K, Pock T, Stollberger R (2011) Second order total generalized variation (tgv) for mri. Magn Reson Med 65(2):480–491
35. Kostadin D, Alessandro F, Vladimir K, Karen E (2007) Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Trans Image Process 16(8):2080–2095
36. Lan X, Roth S, Huttenlocher D, Black MJ (2006) Efficient belief propagation with learned higher-order markov random fields. In: European conference on computer vision. Springer, pp 269–282
37. Liao X, Li K, Yin J (2017) Separable data hiding in encrypted image based on compressive sensing and discrete fourier transform. Multimed Tools Appl 76(20):20739–20753
38. Liao X, Yu Y, Li B, Li Z, Qin Z (2019) A new payload partition strategy in color image steganography. IEEE Transactions on Circuits and Systems for Video Technology
39. Mahmoudi M, Sapiro G (2005) Fast image and video denoising via nonlocal means of similar neighborhoods. IEEE Signal Process Lett 12(12):839–842
40. Mairal J, Bach F, Ponce J, Sapiro G, Zisserman A (2010) Non-local sparse models for image restoration. In: IEEE International conference on computer vision
41. Michael E, Michal A (2006) Image denoising via sparse and redundant representations over learned dictionaries. IEEE Tip 15(12):3736–3745
42. Milovic C, Bilgic B, Zhao B, Acosta-Cabronero J, Tejos C (2018) Fast nonlinear susceptibility inversion with variational regularization. Magn Reson Med 80(2):814–821
43. Mohan J, Krishnaveni V, Guo Y (2014) A survey on the magnetic resonance image denoising methods. Biomed Signal Processing Control 9:56–69
44. Osher S, Wang B, Yin P, Luo X, Barekat F, Pham M, Lin A (2018) Laplacian smoothing gradient descent, arXiv:1806.06317
45. Qin Z, Goldfarb D, Ma S (2015) An alternating direction method for total variation denoising. Optim Methods Softw 30(3):594–615
46. Rudin LI, Osher S, Fatemi E (1992) Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena 60(1-4):259–268
47. Srebro N, Jaakkola T (2003) Weighted low-rank approximations. In: Proceedings of the 20th International Conference on Machine Learning (ICML-03), pp 720–727
48. Tihonov AN (1963) Solution of incorrectly formulated problems and the regularization method. Sov Math 4:1035–1038
49. Wang Y, Yang J, Yin W, Zhang Y (2008) A new alternating minimization algorithm for total variation image reconstruction. SIAM J Imaging Sci 1(3):248–272
50. Wang F, Huang H, Liu J (2019) Variational based mixed noise removal with cnn deep learning regularization. IEEE Transactions on Image Processing
51. Weisheng D, Guangming S, Xin L (2013) Nonlocal image restoration with bilateral variance estimation: a low-rank approach. IEEE Trans Image Process 22(2):700–711
52. Xiao Y, Zeng T, Yu J, Ng MK (2011) Restoration of images corrupted by mixed gaussian-impulse noise via l1–l0 minimization. Pattern Recogn 44(8):1708–1720
53. Xie Y, Gu S, Liu Y, Zuo W, Zhang W, Zhang L (2016) Weighted schatten p-norm minimization for image denoising and background subtraction. IEEE Trans Image Process 25(10):4842–4857
54. Xiong B, Yin Z (2011) A universal denoising framework with a new impulse detector and nonlocal means. IEEE Trans Image Process 21(4):1663–1675
55. Yan M (2013) Restoration of images corrupted by impulse noise and mixed gaussian impulse noise using blind inpainting. SIAM J Imaging Sci 6(3):1227–1245
56. Yang Y, Sun J, Li H, Xu Z (2017) Admm-net: A deep learning approach for compressive sensing mri. corr, arXiv:1705.06869
57. Yang J, Zhang Y, Yin W (2009) An efficient tvl1 algorithm for deblurring multichannel images corrupted by impulsive noise. SIAM J Sci Comput 31(4):2842–2865
58. Zhang Y (2010) An alternating direction algorithm for nonnegative matrix factorization, Technical Report

59. Zhang D, Hu Y, Ye J, Li X, He X (2012) Matrix completion by truncated nuclear norm regularization. In: 2012 IEEE Conference on computer vision and pattern recognition. IEEE, pp 2192–2199
60. Zhang K, Zuo W, Chen Y, Meng D, Zhang L (2017) Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Trans Image Process 26(7):3142–3155
61. Zhang K, Zuo W, Gu S, Zhang L (2017) Learning deep cnn denoiser prior for image restoration. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3929–3938
62. Zhou T, Bhaskar H, Liu F, Yang J, Cai P (2017) Online learning and joint optimization of combined spatial-temporal models for robust visual tracking. Neurocomputing 226:221–237
63. Zhou W, Zhang Y, Liu Y (2017) Matrix-value linear regression for image denoising. DEStech Transactions on Computer Science and Engineering, no. mmsta
64. Zhou T, Liu F, Bhaskar H, Yang J (2018) Robust visual tracking via online discriminative and low-rank dictionary learning. IEEE Trans Cybern 48(9):2643–2655
65. Zhou T, Thung K-H, Liu M, Shen D (2019) Brain-wide genome-wide association study for alzheimer's disease via joint projection learning and sparse regression model. IEEE Trans Biomed Eng 66(1):165–175

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