Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 6339758, 14 pages
https://doi.org/10.1155/2022/6339758

Research Article

A New Boosting Algorithm for Shrinkage Curve Learning

Xiyan Meng[1] **and Fang Zhuang**[2]

[1]*School of Human Science, Enshi Polytechnic, Enshi, Hubei 445000, China*
[2]*School of Mathematics and Statistics, Hubei Minzu University, Enshi, Hubei 445000, China*

Correspondence should be addressed to Xiyan Meng; mengxyyx@163.com and Fang Zhuang; wdfangzhuang@163.com

Received 16 August 2021; Revised 14 March 2022; Accepted 15 March 2022; Published 15 April 2022

Academic Editor: Francesco Lolli

To a large extent, classical boosting denoising algorithms can improve denoising performance. However, these algorithms can only work well when the denoisers are linear. In this paper, we propose a boosting algorithm that can be used for a nonlinear denoiser. We further implement the proposed algorithm into a shrinkage curve learning denoising algorithm, which is a nonlinear denoiser. Concurrently, the convergence of the proposed algorithm is proved. Experimental results indicate that the proposed algorithm is effective and the dependence of the shrinkage curve learning denoising algorithm on training samples has improved. In addition, the proposed algorithm can achieve better performance in terms of visual quality and peak signal-to-noise ratio (PSNR).

1. Introduction

Image denoising is one of the basic steps of many advanced image processing technologies. The purpose of image denoising is to obtain the approximate result of a noise-free image x from the observed image $y = x + v$, so as to minimize the influence of noise v on advanced image processing problems such as computer vision, pattern recognition. In order to achieve this purpose, many scholars have developed a large number of excellent denoising algorithms such as wavelet soft threshold [1], NLM [2], K-SVD [3], and BM3D [4] by using wavelet multi-scale decomposition, spatial domain filtering, frequency domain filtering, sparse redundancy representation, and image patch processing technology, among others [5–8]. It is generally believed that the performance of these traditional single denoising algorithms almost reaches the top of the ceiling [9].

Can traditional denoising performance be further improved? Several scholars have answered this question and proposed several boosting denoising algorithms, such as twining [10], Bregman iterations [11], and SAIF [12]. For example, the main idea of twining technology is to continuously extract the noisefree image components from a residual image in an iterative process, and then add these components back into the estimated image. In the Bregman iterations method, the sum of the residual images obtained in the previous iteration is considered and added to the observed image $\mathbf{y}$ before performing the denoising operation. SAIF [12] is a patch-based boosting denoising algorithm. For each patch, both signal leftovers that reside in the residual image and noise leftovers that reside in the denoised image are considered, and then the algorithm automatically chooses the boosting mechanism.

Unlike the above boosting methods, instead of adding the residual image back into the noisy image, or filtering the previous estimate over and over again, Romano et al. [9] proposed a new boosting denoising algorithm named SOS that strengthens the noise image by adding part of the noisy image to the noise image before executing the next iteration denoiser and subtracting the same part of the outcome; that is

$$\widehat{x}^{k+1} = D_\sigma\left(y + \widehat{x}_k\right) - \widehat{x}^k, \tag{1}$$

where the operator $D_\sigma(\cdot)$ represents the denoising algorithm that depends on the parameter σ and $\widehat{x}^k$ is the kth-iteration denoised image. Further research shows that the SOS algorithm is convergent when $D_\sigma(\cdot)$ satisfies the linearity limitation.

Inspired by SOS, Fang et al. [13] proposed an adaptive boosting denoising algorithm named ABD that plugs in the noise level estimation, which can further improve the performance of SOS, but ABD is still limited by the assumption of linearity.

Although Milanfar [14] pointed out that some denoisers based on patch processing can satisfy the assumption of linearity through Sinkhorn's algorithm [15, 16], and the NLM, KSVD, and BM3D can be boosted by the SOS algorithm, many denoisers, such as threshold denoisers [17, 18], shrinkage curve learning [19], denoisers based on deep learning [20], and others [21, 22], cannot satisfy the condition of linearity; therefore, they cannot be boosted by the existing boosting algorithm.

In fact, based on the sparse domain model of an image, Hel-Or et al. solved the shrinkage curve in a given training set. Specifically, the shrinkage curve must add noise with a given noise level to the clean image in the training set. Then, the model of a noise image shrinking to a clean image is established in the transform domain, and the shrinking coefficient is solved to obtain the shrinking operator. Finally, these shrinkage operators can be used to remove the noise, which is the noisy image with the same noise level. Hel-Or's algorithm overcomes the shortcomings of traditional thresholding algorithms in the entire image-processing process by averaging the local patches of the noise image. However, the performance of the algorithm depends heavily on the images in the given training set. If the structure of the image in the training set is similar to that of the noise image, the algorithm can achieve a better denoising effect [23]. This conclusion is discussed and compared with the experimental results in [19, 23]. Therefore, the denoising algorithm cannot satisfy linearity.

Moreover, the noise level also has an effect on the shrinkage curve learning. When the selected shrinkage curve matches the noise level of the noise image, over-or under-shrinkage can be avoided.

On the basis of previous studies and analysis, in this paper, we focus on a boosting algorithm for nonlinear denoisers, search for methods for boosting the shrinkage curve learning denoising algorithm and discuss the convergence condition of the boosting algorithm. The main contributions to this paper are the following:

(1) To solve the problem that the performance of a shrinkage curve learning denoising algorithm depends on the training samples and noise level, a boosting algorithm based on shrinkage curve learning denoising is proposed. In the proposed algorithm, the training samples come from the noise image itself, and the threshold of the shrinkage curve is automatically calculated by the noise level estimation algorithm. The algorithm proposed in this paper not only does not depend on the external image but can also be used to devise a boosting denoising algorithm with a nonlinear property.

(2) Theoretical analysis shows that the proposed boosting algorithm is convergent when the noise level has a finite growth property, and the effectiveness of the algorithm is verified by experiments.

The rest of this paper is organized as follows: in Section 2, the principle of denoising based on shrinkage curve learning is briefly introduced. In Section 3, the boosting algorithm based on shrinkage curve learning and the basic flow of the algorithm are given, and the convergence of the algorithm is analyzed and proved. In Section 4, the experimental results of the algorithm are presented and analyzed. In Section 5, conclusions and future work are given.

2. Previous Related Work

2.1. Patch-Based Shrinkage Method. Shrinkage curve learning is a denoising algorithm based on image patch processing. In such an algorithm, the noisy image y is divided into patches of $N = w \times w$, and the pixels in these patches are organized by column vectors. According to Elad's amputation, these column vectors satisfy the sparse domain model [23]; that is, there exists a transformation $A \in \mathbb{R}^{N\times m}$, for each vector $q \in \mathbb{R}^m$, it can be presented $\|Aq - p\|_2 \le \varepsilon$, and $\|q\|_0 = k_0$, where $\|q\|_0$ denotes the number of nonzero entries of q.

When the sparse domain model is used to denoise an image, a natural choice is to process each patch individually and then splice them together; in that way, artificial marks may be formed on the edge. To avoid artificial marks on the edge when splicing between patches, it is necessary to average the results of overlapping patches. In practical calculations, the transformation is given, such as the 2D DCT unitary matrix or the 2D DWT matrix.

The following steps generally comprise the entire image-denoising process.

(1) Each pixel in y is regarded as the center of a patch of $w \times w$ size; the patch is denoted as p_{ij}.

(2) Using the threshold algorithm, the following steps are performed for each patch p_{ij} for denoising.

  (i) calculates $\widetilde{q}_{ij} = A^T p_{ij}$.

  (ii) the vector obtained from the pair of $\widetilde{q}_{ij}$, using the preset threshold for threshold operation, $\widetilde{q}_{ij}$ can be obtained.

  (iii) calculates $\widehat{p}_{ij} = A^T \widetilde{q}_{ij}$, obtains the denoising result of each patch.

(3) The fusion result is obtained by averaging all the denoised patches.

2.2. Shrinkage Curve Learning. The main process of the algorithm presented in the preceding subsection can be expressed by the following formula:

$$\widehat{p}_{ij} = A\widehat{q}_{ij} = AS\left\{A^T p_{ij}\right\}, \tag{2}$$

where S is a shrinkage operator. Hel-Or et al. [19] solved S as follows:

Gaussian noise with zero mean value and a given variance of σ^2 is added to the images in the training set. A clean image and its corresponding noisy image are used to construct training samples x_{train}. Let $\{p_k^0\}_{k=1}^M$ and $\{p_k\}_{k=1}^M$ denote clean patches and their noisy version, respectively. We use

M pairs of patches, $\{p_k^0, p_k\}_{k=1}^M$, to determine the shrink rule, which is used to replace the hard-threshold algorithm and obtain the optimal threshold parameter set.

To this end, the penalty function is set as follows:

$$F(S) = \sum_{k=1}^{M} \left\| p_k^0 - AS\left\{A^T p_k\right\}\right\|_2^2. \tag{3}$$

From (3), the penalty function is a series of shrink operators. The purpose of the approach of Hel-Or et al. is to use these shrink operators to shrink the noisy image of the above M patches to obtain the closest result to the clean image patches.

For this reason, it is supposed that the shrink operators for each element in the input vector are different, and then the shrink processing for any vector $u \in \mathbb{R}^m$ is

$$S\{u\} = \begin{bmatrix} S_1\{u[1]\} \\ S_2\{u[2]\} \\ \vdots \\ S_m\{u[m]\} \end{bmatrix}. \tag{4}$$

Letting $u_k = A^T p_k$, and inserting it into (3), the penalty function is then

$$\begin{aligned} F(S_1, S_2, \ldots, S_m) &= \sum_{k=1}^{M} \left\| p_k^0 - AS\left\{A^T p_k\right\}\right\|_2^2 \\ &= \sum_{k=1}^{M} \left\| p_k^0 - \sum_{i=1}^{m} a_i S_i\{u_k[i]\}\right\|_2^2. \end{aligned} \tag{5}$$

Because $\mathbf{A}$ is a unitary matrix, then $\mathbf{u}_k^0 = \mathbf{A}^T \mathbf{p}_k^0$, and

$$\begin{aligned} F(S_1, S_2, \ldots, S_m) &= \sum_{k=1}^{M} \left\| p_k^0 - AS\left\{A^T p_k\right\}\right\|_2^2 \\ &= \sum_{k=1}^{M} \left\| A^T p_k^0 - S\left\{A^T p_k\right\}\right\|_2^2 \\ &= \sum_{k=1}^{M} \left\| u_k^0 - S\{u_k\}\right\|_2^2 \\ &= \sum_{k=1}^{m} \sum_{k=1}^{M} \left(u_k^0[i] - S_i\{u_k[i]\}\right)^2. \end{aligned} \tag{6}$$

Equation (6) implies that m shrinkage curves can be obtained independently. According to Hel-Or's method, the following polynomial contraction was used:

$$S_i\{u\} = \sum_{j=0}^{J} c_i[j] u^j, \tag{7}$$

where $c_i[j]$ is the coefficient of a polynomial, and the least-squares method was used to calculate the optimal solution of the shrinkage curve $\{c_i[j]\}_{j=0}^J$.

Therefore, inserting (7) into (5) obtains

$$\begin{aligned} F(S_1, S_2, \ldots, S_m) &= \sum_{k=1}^{M} \left\| p_k^0 - \sum_{i=1}^{m} a_i S_i\{u_k[i]\}\right\|_2^2 \\ &= \sum_{i=1}^{M} \left\| p_k^0 - \sum_{i=1}^{m} a_i \sum_{j=0}^{J} c_j[j] u_k[i]^j\right\|_2^2. \end{aligned} \tag{8}$$

Letting $c \in \mathbb{R}^{mJ}$, it contains all of the $\{c_i[j]\}_{j=0}^J$ sequences, $i = 1, 2, \ldots, m$, that is

$$\begin{aligned} c^T = [&c_1[0], c_1[1], \ldots, c_1[J], c_2[0], c_2 \\ &\cdot [1], \ldots c_2[J], c_m[0], \ldots c_m[J],]. \end{aligned} \tag{9}$$

Similarly, defines a patch diagonal matrix $U_k \in R^{m \times mJ}$ that contains M patches

$$U_k = \begin{bmatrix} b_k[1] & 0 & \cdots & 0 \\ 0 & b_k[2] & \cdots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & b_k[m] & \end{bmatrix}. \tag{10}$$

The size of each patch $b_k[i]$ is $1 \times (J+1)$, and the value is

$$b_k[i] = \left[u_k[i]^0, u_k[i]^2, \ldots u_k[i]^J\right]. \tag{11}$$

Using the above denotation, the minimum penalty function is expressed as

$$F(c) = \sum_{k=1}^{M} \left\| p_k^0 - AU_k c\right\|_2^2. \tag{12}$$

Let $(\partial F(c)/\partial c) = 0$, that is

$$\frac{\partial F(c)}{\partial c} = \sum_{k=1}^{M} U^T A^T \left(AU_k c - p_k^0\right) = 0. \tag{13}$$

The solution of c can be obtained from (13) as

$$c_{\text{opt}} = \left[\sum_{k=1}^{M} U_k^T A^T A U_k\right]^{-1} \sum_{k=1}^{M} U_k^T A^T p_k^0, \tag{14}$$

which is also the optimal parameter of the shrinkage curve. Therefore, a family of shrinkage curves in the training set $\{p_k^0, p_k\}_{k=1}^M$ are obtained, which can be used for new images incurring interference from the additive noise with the same noise level of σ.

3. Boosting Algorithm for Shrinkage Curve Learning

Different from the linear denoiser used in the boosting algorithm in [9, 13], a denoiser composed using a shrinkage curve family is nonlinear and is related to the training samples. To determine how to denoise with and boost this

kind of denoiser requires some changes in the form of the denoising and denoising algorithm in [9, 13].

3.1. Proposed Algorithm. In the preceding section, the optimal parameter set of the shrinkage curve comes from the training set. Therefore, the coefficient c_{opt} in (14) is related to two factors, namely, the training sample x_{train} and noise level σ. Denoting c_{opt} as $T(x_{\text{train}}, \sigma)$, (14) can then be rewritten as

$$T(x_{\text{train}}, \sigma) = \left[\sum_{k=1}^{M} U_k^T A^T A U_k\right]^{-1} \sum_{k=1}^{M} U_k^T A^T p_k^0, \tag{15}$$

In the transform domain, the high-frequency components can be effectively filtered out by using the shrinkage curve determined by the coefficient c_{opt}. Then, we obtain a denoiser that is related to $T(x_{\text{train}}, \sigma)$, this denoiser is denoted as $D_{x_{\text{train}},\sigma}$, where the subscript shows that the denoiser is related to the training set x_{train} and the given noise level σ. The boosting algorithm of the denoiser is discussed as follows, and it is different from the linear denoiser reported by Fang [13]. The denoiser $D_{x_{\text{train}},\sigma}$ is a comprehensive denoiser composed of a series of nonlinear shrinkage operators and is a typical nonlinear denoiser.

To boost the effect of $D_{x_{\text{train}},\sigma}$ and overcome the dependence of $D_{x_{\text{train}},\sigma}$ on an external database and noise level, we use the following steps to boost $D_{x_{\text{train}},\sigma}$:

(1) The noise level σ of noisy image y is estimated.

(2) c_{opt} and D_σ are initialized, using the constant value image to train the shrinkage curve; that is $c_{\text{opt}} = T(0_{\text{train}}, \sigma)$, and then D_{σ_0} and the initial denoised image can be calculated.

(3) The denoised image is added to the input noise image, and the noise level of the accumulated image is calculated. A new denoiser is obtained by training the shrinkage operator with the denoised image.

(4) Step (3) is repeated until the denoising result converges.

The above process is described by the following equation:

$$\begin{cases} \widehat{y}_k = \widehat{y}_{k-1} + \rho \widehat{x}_{k-1}, \\ \\ \widehat{x}_k = \dfrac{D_{c\sigma_k}(y_k)}{1 + (k-1)\rho}, \end{cases} \tag{16}$$

where c controls the denoising strength, ρ controls the return ratio of clean images, and σ_k is the noise level of $\widehat{y}_k$. Many algorithms estimate the latter, e.g., PCA [24], WT [25], among others [26, 27]. In this paper, the noise level is estimated by the algorithm reported in [27]. The specific algorithm corresponding to the process is shown as algorithm 1.

3.2. Convergence Analysis of Proposed Algorithm. The convergence of the proposed algorithm 1 is discussed here. First, the definition of a shrinkage denoiser is given as follows:

Definition 1 (Shrinkage denoiser). The denoiser $D_\sigma(x)$ with denoising strength σ is called a shrinkage denoiser if it satisfies

$$\left\|D_\sigma(x)\right\|_2^2 \le \|x\|_2^2. \tag{17}$$

Definition 2 (Bounded denoiser) (see [28]). A bounded denoiser with a parameter σ as a function $D_\sigma(x)$: $R^n \longrightarrow R^n$ such that, for any input $x \in R^n$,

$$\left\|D_\sigma(x) - x\right\|_2^2 \le nG\sigma^2, \tag{18}$$

for some universal constant G independent of n and σ.

From the analysis in Section 2.2, we can see that D_σ is the result of a finite number of shrinkage operators.

Proposition 1 (see [27]). *There exists a constant M satisfying $\|\sigma_k\| \le M$.*

Proposition 2. *The sequence $\{\sigma_k/k\}$ convergence and*

$$\lim_{k \longrightarrow \infty} \frac{\sigma_k}{k} = 0. \tag{19}$$

Proof. From Proposition 1, there exists M, s.t. $\|\sigma_k\| \le M$. Dividing both sides by k, and letting $k \longrightarrow \infty$, we have $\lim_{k \longrightarrow \infty} \sigma_k/k = 0$.

Based on these definitions and properties, the following properties can be obtained: □

Proposition 3. *For a given positive integer k, the following inequality holds: $\|y_k\|_2 \le (1 + k\rho)\|y_0\|_2$ and $\|x_k\|_2 \le \|y_0\|_2$.*

Proof. First, we prove that $\|y_k\|_2 \le (1 + k\rho)\|y_0\|_2$. From (16), we have

$$\begin{aligned} \left\|y_k\right\|_2 = \left\|y_{k-1} + \rho x_k\right\|_2 &= \left\|y_{k-1} + \rho \frac{D_{\sigma_k}(y_{k-1})}{1 + (k-1)\rho}\right\|_2 \\ &\le \left\|y_{k-1}\right\|_2 + \frac{\rho}{1 + (k-1)\rho}\left\|D_{\sigma_k}(y_{k-1})\right\|_2. \end{aligned} \tag{20}$$

For D_σ is a shrinkage denoiser, using the inequality in Definition 1, we have

$$\left\|D_{\sigma_k}(y_{k-1})\right\|_2 \le \left\|y_{k-1}\right\|_2. \tag{21}$$

We take equations (20) to equation (19), after some simplifications, we obtain

Input: noisy image y.
Output: denoised image $\widehat{x}_k$.
Initialization: $y_0 = y$, the noise level σ_0 of the image y. $\widehat{x}_0 = 0$, maximum iteration K, $k = 0$.
Main iteration: while $k < K$, do
(1) : $k = k + 1$.
(2) : using algorithm1 in [27] to compute the noise level of the image y_{k-1}, denote it as σ_{k-1}.
(3) : $MF_k = T(\widehat{x}_{k-1}, c\sigma_{k-1})$.
(4) : $\widehat{u}_k = D_{MF_k}(y_{k-1})$
(5) : $\widehat{x}_k = \widehat{u}_k/[1 + (k-1)\rho]$.
(6) : $y_k = y_{k-1} + \rho\widehat{x}_k$.
Return: denoised image $\widehat{x}_k$.

Algorithm 1: adaptive shrinkage curve learning denoising algorithm.

$$\|y_k\|_2 = \frac{1+k\rho}{1+(k-1)\rho}\|y_{k-1}\|_2. \tag{22}$$

Then the recurrence law is obtained as follows:

$$\begin{aligned}\|y_k\|_2 &\le \frac{1+k\rho}{1+(k-1)\rho}\frac{1+(k-1)\rho}{1+(k-2)\rho}\|y_{k-2}\|_2 \le \cdots \\ &\le \frac{1+k\rho}{1+(k-1)\rho}\frac{1+(k-1)\rho}{1+(k-2)\rho}\frac{1+(k-2)\rho}{1+(k-3)\rho}\cdots\frac{1+\rho}{1}\|y_0\|_2.\end{aligned} \tag{23}$$

After further simplification, we obtain

$$\|y_k\|_2 \le (1+k\rho)\|y_0\|_2. \tag{24}$$

Next, we prove that: $\|x_k\|_2 \le \|y_0\|_2$.
Because

$$\|x_k\|_2 = \left\|\frac{D_{\sigma_k}(y_{k-1})}{1+(k-1)\rho}\right\|_2 \le \frac{\|D_{\sigma_k}(y_{k-1})\|_2}{1+(k-1)\rho} \le \frac{\|y_{k-1}\|_2}{1+(k-1)\rho}. \tag{25}$$

Equation (24) is inserted (25), we have

$$\|x_k\|_2 \le \frac{1+(k-1)\rho}{1+(k-1)\rho}\|y_0\|_2 = \|y_0\|_2. \tag{26}$$

This completes the proof. □

Theorem 1. *If $D_\sigma(\cdot)$ is a bounded and shrinkage denoiser, then the solution sequence obtained from the iterative equation (15) in algorithm 1 is convergent; that is $\|\widehat{x}_{k+1} - \widehat{x}_k\|_2 \longrightarrow 0$ when $k \longrightarrow \infty$.*

Proof. From equation (15), we have

$$\begin{aligned}&\|x_{k+1} - x_k\|_2 \\ &= \left\|\frac{D_{\sigma_{k+1}}(y_k)}{1+k\rho} - \frac{D_{\sigma_k}(y_{k-1})}{1+(k-1)\rho}\right\|_2 \\ &\le \left\|\frac{D_{\sigma_{k+1}}(y_k) - y_k + y_k}{1+k\rho} - \frac{D_{\sigma_k}(y_{k-1}) - y_{k-1} + y_{k-1}}{1+(k-1)\rho}\right\|_2 \\ &\le \left\|\frac{D_{\sigma_{k+1}}(y_k) - y_k}{1+k\rho} + \frac{y_k}{1+k\rho} - \frac{D_{\sigma_k}(y_{k-1}) - y_{k-1}}{1+(k-1)\rho} - \frac{y_{k-1}}{1+(k-1)\rho}\right\|_2.\end{aligned} \tag{27}$$

Using trigonometric inequality, we have

$$\begin{aligned}\|x_{k+1} - x_k\|_2 &\le \left\|\frac{D_{\sigma_{k+1}}(y_k) - y_k}{1+k\rho}\right\|_2 + \left\|\frac{D_{\sigma_k}(y_{k-1}) - y_{k-1}}{1+(k-1)\rho}\right\|_2 \\ &+ \left\|\frac{y_k}{1+k\rho} - \frac{y_{k-1}}{1+(k-1)\rho}\right\|_2.\end{aligned} \tag{28}$$

For the first and second items on the right-hand side of (28), using the definition of a bounded denoiser given in Section 3.2, we obtain

$$\begin{aligned}&\left\|\frac{D_{\sigma_{k+1}}(y_k) - y_k}{1+k\rho}\right\|_2 + \left\|\frac{D_{\sigma_k}(y_{k-1}) - y_{k-1}}{1+(k-1)\rho}\right\|_2 \\ &\le \frac{\sqrt{NG}\sigma_{k+1}}{1+k\rho} + \frac{\sqrt{NG}\sigma_k}{1+(k-1)\rho}.\end{aligned} \tag{29}$$

Inserting $\widehat{y}_k = \widehat{y}_{k-1} + \lambda\widehat{x}_{k-1}$ into the last item on the right-hand side of (28), we obtain

$$\begin{aligned}\left\|\frac{y_k}{1+k\rho} - \frac{y_{k-1}}{1+(k-1)\rho}\right\|_2 &= \left\|\frac{y_{k-1} + \rho x_k}{1+k\rho} - \frac{y_{k-1}}{1+(k-1)\rho}\right\|_2 \\ &= \left\|\left(\frac{1}{1+k\rho} - \frac{1}{1+(k-1)\rho}\right)y_{k-1} + \frac{\rho x_k}{1+k\rho}\right\|_2 \\ &\le \frac{\rho\|y_{k-1}\|_2}{(1+k\rho)[1+(k-1)\rho]} + \frac{\rho\|x_k\|_2}{1+k\rho}.\end{aligned} \tag{30}$$

Using Proposition 3, we have

$$\begin{aligned}\left\|\frac{y_k}{1+k\rho} - \frac{y_{k-1}}{1+(k-1)\rho}\right\|_2 &\le \frac{\rho[1+(k-1)\rho]\|y_0\|_2}{(1+k\rho)[1+(k-1)\rho]} + \frac{\rho\|x_0\|_2}{1+k\rho} \\ &= \frac{\rho}{1+k\rho}\left(\|x_0\|_2 + \|y_0\|_2\right).\end{aligned} \tag{31}$$

Inserting (29) and (31) into (28), we have

$$\left\|x_{k+1}-x_k\right\|_2 \leq \frac{\sqrt{NG}\sigma_{k+1}}{1+k\rho}+\frac{\sqrt{NG}\sigma_k}{1+(k-1)\rho} + \frac{\rho}{1+k\rho}\left(\left\|x_0\right\|_2+\left\|y_0\right\|_2\right). \tag{32}$$

Because $\|x_0\|_2$ and $\|y_0\|_2$ are bounded, we utilize Proposition 2, take the limit on both sides of equation (32), and obtain $\|x_{k+1}-x_k\|_2 \longrightarrow 0$ when $k \longrightarrow \infty$. □

4. Experimental Results and Analysis

The values of the two parameters of algorithm 1 were determined by trial and error, and the properties of Proposition 2 were verified by experiments. All the methods are implemented in MATLAB (16 GB RAM, 3.00 GHz CPU, and an Intel(R) Core(TM) I7-9700 processor).

4.1. Parameter Setting. In proposed algorithm 1, the parameter ρ is used to control the weight of the clean image to return to the next iteration. The larger the value of ρ, the higher the signal-to-noise ratio (SNR) of y_k from the iteration $y_k = y_{k-1} + \rho x_{k-1}$. c is used to control the intensity of noise added to the training sample. The larger the value of c, the greater the noise level added to the training sample, and the stronger the denoising ability of the trained shrinkage curve to the high noise image. On the contrary, the smaller the value of c, the more suitable the trained shrinkage curve is for low noise image denoising. In this paper, c and ρ are selected as follows: we limit c to the interval $[0, 2]$, and ρ to the interval $[0.2, 1.2]$. We take the point in the two intervals in steps of 0.1 to obtain the parameter pair $\{\rho_i, c_i\}$. The standard test image (as shown in Figure 1) is added with Gaussian noise with a mean value of 0 and variance of σ^2 to obtain the noise image. These parameter pairs are used in $\{\rho_i, c_i\}$ into algorithm 1 to denoise the noise image. Figure 2 shows the relationship among ρ, c, and PSNR. It can be seen that the PSNR value is the highest and the denoising effect is the best when $\rho = 0.6$ and $c = 1.02$. In the experiments described as follows, the parameter values in algorithm 1 were set as $\rho = 0.6$ and $c = 1.02$.

4.2. Verification of Shrinkage Properties. Proposition 2 plays a key role in the process of proving the convergence of algorithm 1. The convergence of the sequence $\{\sigma_k/k\}$ generated in algorithm 1 is verified in this subsection. For all images and noise levels, the sequence $\{\sigma_k/k\}$ converges to 0 just like the curve in Figure 3(a), and no difference exists. As can be seen from the figure, with increasing iteration number k, the sequence $\{\sigma_k/k\}$ converges rapidly to 0. These experimental results further verify the validity of Proposition 2. Correspondingly, considering the change in PSNR value after each step of the iteration, the results are given in Figure 3(b). It is found that the PSNR value obtained by algorithm 1 converges to a stable state quickly, and it is also shown that algorithm 1 only needs a few iterations to achieve a good denoising effect. Therefore, the proposed algorithm can be terminated directly by truncating the number of iterations, thus avoiding the extra operation of using blind image quality evaluation to determine the optimal iteration number. In the experiments discussed later in this paper, the number of iterations was set to 9.

4.3. Comparison of Shrinkage Curves. The training of shrinkage curves plays an important role in the proposed algorithm 1. In Figure 4 the differences between the original algorithm in [19] and algorithm 1 proposed in this paper are compared. In this experiment, the size of the image patch is set to 8×8. Therefore, a total of 64 shrinkage curves can be obtained, the numbers of which are denoted from 1 to 64. Owing to page-length limitations and for the convenience of comparison, only four curves in the lower right corner, numbered 55, 56, 63, and 64, are compared here. The reason for choosing these four curves is that the high-frequency components of the image patch are mainly concentrated in this part after the DCT transformation. Figure 4 shows the shrinkage curves used by different algorithms to denoise the cameraman image with the noise level of 15.

Subgraph (a) of Figure 4 shows the shrinkage curve used in [19], while subgraphs (c)-(d) show the shrinkage curves used in algorithm 1 for two, five, and nine iterations. The shrinkage curves of the two algorithms hardly operate on the DC part of the upper left-hand corner, but for the AC part, the curve used in [19] shrinks the high-frequency part, and the closer the shrinkage curve is to the high-frequency part, the stronger the shrinkage curve. The curve used in each iteration of the proposed algorithm 1 has the same trend as the curve in [19], but in the low-frequency part, the curve shrinkage in algorithm 1 is stronger than that in [19]. We note that the curve numbered 55 in Figure 4(a), the shrinkage in [19] is weak, while the shrinkage curve numbered 55 in Figures 4(b)–4(d) increased gradually using the proposed algorithm.

A similar conclusion can be obtained for curve 56. This shows that the denoising performance of the algorithm is continuously boosted in the successive iteration process, and the purpose of boosting and denoising is achieved. In terms of the contracted frequency band, e.g., curve 64, the frequency band contracted in [19] is wider. In the iteration of the proposed algorithm, for example, the frequency band of curve 64's contraction from Figure 4(b) to 4(d) is narrower, that is to say, the proposed method achieves a better denoising effect by several iterations of weaker but finer shrinkage.

4.4. Comparison of Denoising Results. The complexity and the actual denoising effect of algorithm 1 are discussed in this subsection.

The complexity of algorithm 1 is mainly produced by three processes: shrinkage curve learning, denoising, and noise level estimation. When the image size is $m \times n$ and the patch size is $w \times w$, the computational complexity of the shrinkage curve learning and denoising processes is $O(w^2(m-w+1)(n-w+1))$, and the computational complexity of noise level estimation is $O(w^2(m-w+1)(n-w+1))$. Therefore, the overall

Figure 1: Test images used are Lena, Barbara, Boat, Couple, Fingerprint, Hill, House, Man, Montage, and Peppers.

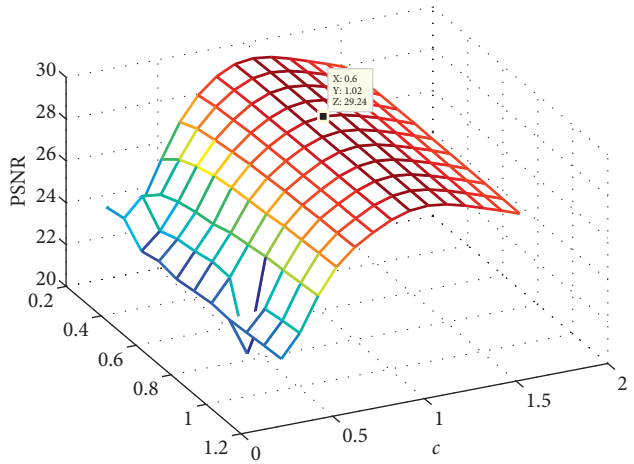

Figure 2: ρ, c, and corresponding PSNR values. In this example of algorithm 1 parameter selection, the test image is Barbara with size 512×512, and the simulated noise is Gaussian noise with a mean value of zero and a standard deviation of 20. This figure shows the relationship between the different parameters ρ, c, and PSNR.

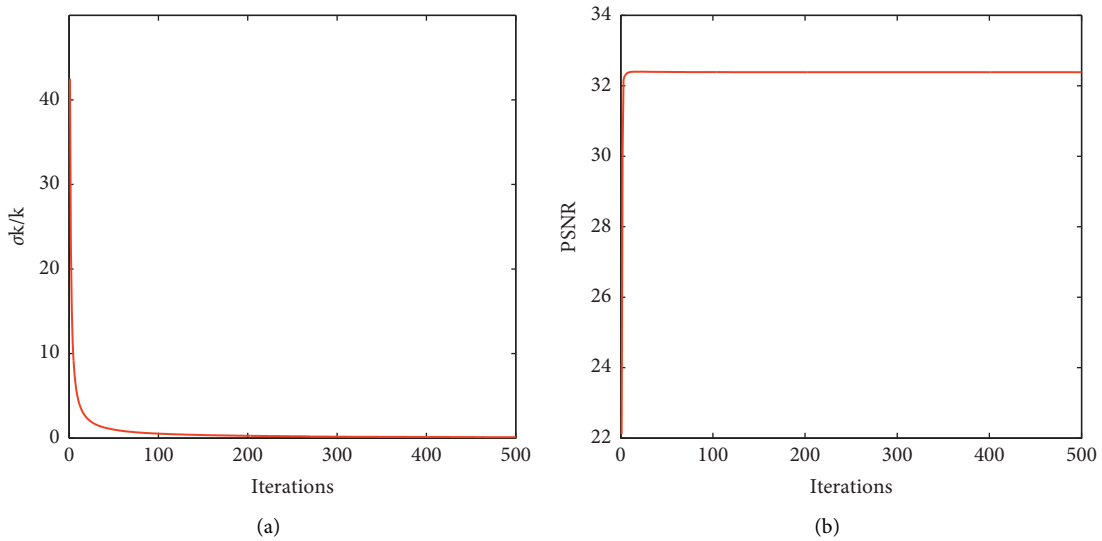

Figure 3: Relationship among sequence $\{\sigma_k/k\}$, PSNR, and iteration number. The image corresponding to this example is a fingerprint with a size of 512×512. The simulated noise is Gaussian noise with zero mean and 10 standard deviations. (a) The sequence $\{\sigma_k/k\}$ converges rapidly to 0 when k increases. (b) PSNR tends to be stable with increasing of k.

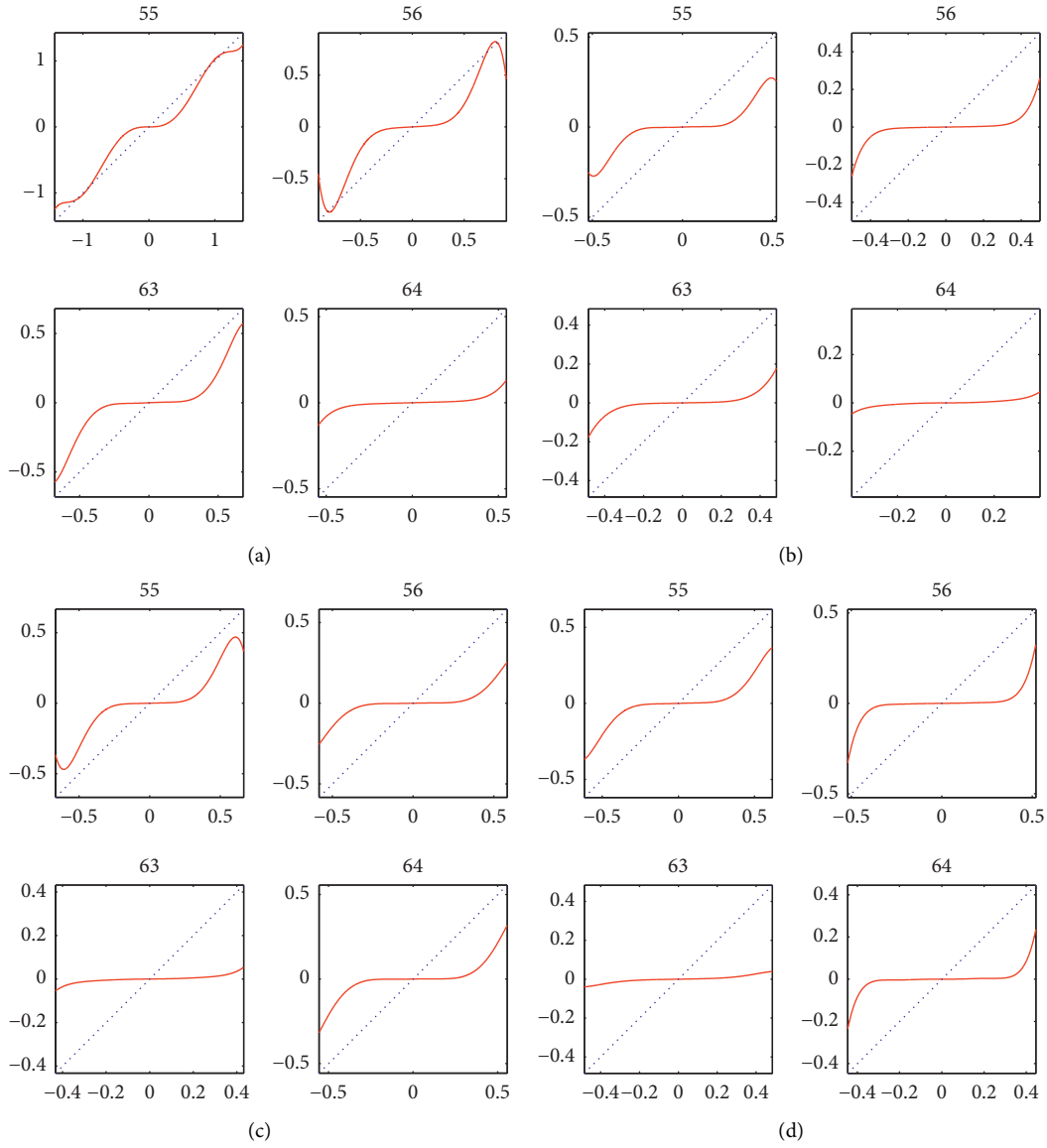

FIGURE 4: The corresponding image of this example is the cameraman image with a size of 256 × 256, and the simulated noise is denoising of a Gaussian noise image with the mean value of zero and standard deviation of 15. Original shrinkage curves, and (b–d) are the shrinkage curves used in two, five, and nine iterations of algorithm 1. (a) Original shrinkage curve. (b) Update 2. (c) Update 5. (d) Update 9.

computational complexity of algorithm 1 is $3kO(w^2(m-w+1)(n-w+1))$, where k is the number of iterations.

The effectiveness of algorithm 1 is illustrated further using the results of three groups of comparative experiments. The first group of experiments used 10 images, which are shown in Figure 1. The second used five texture images from the MeasTex [29] database, and the third used 25 color images from the Tid2008 [30] database.

First, using the test image in Figure 1, Gaussian noise with noise levels of 5, 10, 15, 20, 25, and 50 was added to each clean image, and then the noisy image was obtained. We used the denoising algorithm advanced in [19] and carried out image denoising experiments for three cases: known noise level, estimated noise level, and algorithm 1. To avoid errors, the average value of many experiments was taken as the final result, and the results (PSNR values of noisy images under different noise levels) are presented in Table 1. In the table, the value of the column heading σ_0 represents the denoising result using the algorithm reported in [19] when the noise level is known. The numerical value of the column headed of $\widehat{\sigma}$ represents the result of image denoising when the noise level is the estimated result. The denoising performance of the two algorithms is sensitive to the training image. When the structure of the training sample is similar to the noise image, the denoising effect is the best. Therefore, for the two algorithms, the denoised image itself was used as the training sample to achieve the best denoising effect and ensure the fairness of the comparative experiments. The value of the column headed "Our" represents the result of Algorithm 1. The column labeled "impv" represents the difference between the results obtained by algorithm 1 and

Table 1: Comparison of several denoising results under different noise levels. Test images are from Figure 1.

Image	$\sigma = 5$				$\sigma = 10$				$\sigma = 15$			
	σ_0	$\widehat{\sigma}$	Ours	Impv	σ_0	$\widehat{\sigma}$	Ours	Impv	σ_0	$\widehat{\sigma}$	Ours	Impv
Cameraman	35.91	35.91	36.19	0.28	31.60	31.60	32.01	0.41	29.49	29.50	29.89	0.40
Lena	36.82	36.96	37.91	1.09	33.29	33.35	34.65	1.35	31.64	31.65	32.81	1.17
Barbara	36.57	36.63	36.87	0.30	32.54	32.56	32.90	0.36	30.45	30.46	30.85	0.39
Boat	36.33	36.36	36.57	0.24	32.48	32.48	33.04	0.56	30.58	30.57	31.11	0.53
Couple	35.18	34.83	35.47	0.29	30.44	3dsl0.28	30.99	0.54	28.06	28.00	28.59	0.53
Fingerprint	36.10	36.03	36.67	0.57	31.86	31.86	32.43	0.57	29.59	29.60	30.10	0.51
Hill	35.06	34.89	35.44	0.37	30.59	30.48	30.96	0.36	28.55	28.51	28.82	0.27
House	36.55	36.57	38.34	1.79	32.88	32.93	34.71	1.83	31.20	31.23	32.74	1.54
Man	35.03	34.77	35.31	0.28	30.33	30.21	30.72	0.39	28.10	28.01	28.24	0.14
Montage	36.75	36.76	37.06	0.32	32.82	32.83	33.20	0.38	30.78	30.79	31.20	0.42
Peppers	36.26	36.31	36.96	0.70	32.30	32.33	33.07	0.78	30.33	30.34	31.00	0.68
Image	$\sigma = 20$				$\sigma = 25$				$\sigma = 50$			
	σ_0	$\widehat{\sigma}$	Ours	Impv	σ_0	$\widehat{\sigma}$	Ours	Impv	σ_0	$\widehat{\sigma}$	Ours	Impv
Cameraman	28.10	28.12	28.50	0.40	27.04	27.06	27.45	0.40	23.45	23.41	24.34	0.89
Lena	30.54	30.53	31.50	0.96	29.68	29.64	30.46	0.78	26.86	26.82	27.16	0.30
Barbara	29.02	29.02	29.36	0.34	27.91	27.90	28.15	0.24	24.46	24.48	24.10	-0.35
Boat	29.31	29.30	29.75	0.44	28.34	28.33	28.71	0.37	25.34	25.33	25.64	0.30
Couple	26.56	26.53	27.04	0.48	25.50	25.48	25.92	0.42	22.72	22.71	22.84	0.11
Fingerprint	28.01	28.02	28.54	0.53	26.77	26.79	27.35	0.58	22.70	22.75	23.55	0.86
Hill	27.30	27.30	27.50	0.21	26.42	26.44	26.57	0.15	24.06	24.05	24.17	0.12
House	30.08	30.10	31.41	1.33	29.22	29.23	30.41	1.19	26.47	26.46	27.43	0.96
Man	26.74	26.69	26.70	-0.04	25.79	25.76	25.68	-0.11	23.28	23.27	23.20	-0.07
Montage	29.36	29.37	29.84	0.49	28.21	28.23	28.80	0.59	24.39	24.30	25.49	1.10
Peppers	29.00	29.01	29.60	0.60	27.96	27.97	28.52	0.56	24.51	24.46	25.09	0.58

those obtained using the algorithm in [19]. It can be seen from the table that algorithm 1 can achieve the highest PSNR value for almost all images and all noise levels. Compared with other algorithms, when the proposed algorithm 1 is used to process the house image with a noise level of 10, the PSNR value is improved to a maximum of 1.86 dB.

From the actual processing effect, the proposed algorithm 1 also has the advantage of preserving image details. As shown in Figure 5, the first column represents the original image. The second column represents the noisy image, and its noise level is shown in the figure. The third column is the result obtained by using the algorithm presented in [19] with an accurate noise level. The fourth column lists the results that use the estimated noise level. The last column is the result of algorithm 1. It can be seen from Figure 5, for example, that the edge of a hat in the Lena image, the ridgeline in the house image, and the pepper's contour in the pepper image is clearer than when processed by other algorithms. In the fingerprint image, the overall texture of the denoised image is clearer. In the montage image, even if high-level noise is added, the edge line of the montage image is still clearer.

Next, we considered the influence of texture images on the proposed algorithm by using five images from the MeasTex [29] database, i.e., fabric, food, leaves, water, and wood. All five images contain rich texture features. For each clean image, Gaussian noise with noise levels of 5, 10, 15, 20, 25, and 50 was added. The results of image denoising using algorithm 1 are shown in Table 2, from which it can be seen that the greatest improvement of algorithm 1 compared with the algorithm given in [19] is 4.28 dB, and the average improvement is 1.49 dB. Thus, compared with the algorithm reported in [19], the proposed algorithm 1 significantly improved the denoising of richly textured images.

Finally, the influence of color images in natural scenes on the algorithm was considered using 25 images from the Tid2008 [30] database, that contained different scenes and characters; Gaussian noise was added to each image. The results of image denoising using algorithm 1 are shown in Table 3, from which it can be seen that algorithm 1 significantly improved the denoising effect on all images selected from the Tid2008 database. Moreover, compared with the algorithm presented in [19], the proposed algorithm 1 was able to better maintain the details of color images. This can be seen from the comparison in Figures 6(c) and 6(d). Note further that the edge of the hat in Figure 6(e) is closer to the edge of the hat in Figure 6(a) without noise, and there is no jagged edge. Compared with Figures 6(h) and 6(i), the boundaries between water and beach and between forest and sky in Figure 6(j) are clearer. Similarly, compared with Figures 6(m) and 6(n), the parrot image in Figure 6(o) maintains a better texture and edge. It can therefore be seen that algorithm 1 has a better noise-removal effect for images of various natural scenes and has a better ability to maintain details than the algorithm presented in [19].

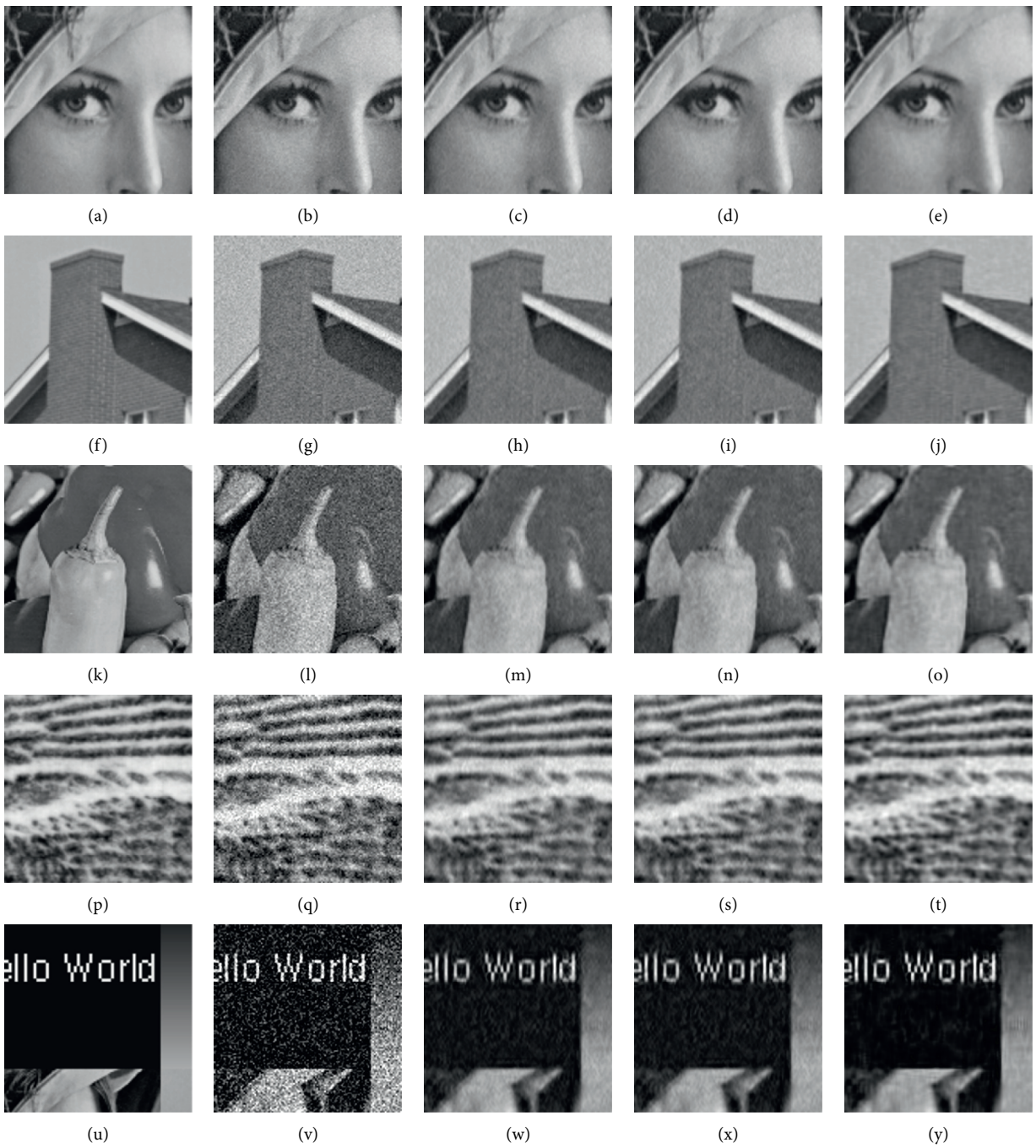

Figure 5: Visual and PSNR comparisons of several algorithms of 120×120 cropped region from a noisy image. The first column is a noisefree image, the second column is a noisy image, whose corresponding noise level is below the image, the third and fourth columns are the results of curve denoising trained by using accurate and estimated noise levels, respectively, and the fifth column shows the denoising results of the proposed algorithm. (a) Lena. (b) $\sigma = 5$. (c) $\sigma_0 = 36.82$. (d) $\widehat{\sigma} = 36{:}96$. (e) Ours = 37.91. (f) House. (g) $\sigma = 10$. (h) $\sigma_0 = 32.88$. (i) $\widehat{\sigma} = 32.93$. (j) Ours = 34.71. (k) Peppers. (l) $\sigma = 20$. (m) $\sigma_0 = 29.00$. (n)$\widehat{\sigma} = 29.01$. (o) Ours = 29.60. (p) Fingerprint. (q) $\sigma = 25$. (r) $\sigma_0 = 26.77$. (s) $\widehat{\sigma} = 26.79$. (t) Ours = 27.35. (u) Montage. (v) $\sigma = 50$. (w) $\sigma_0 = 24.39$. (x)$\widehat{\sigma} = 24.30$. (y) Ours = 25.49.

Table 2: Comparison of several denoising results under different noise levels. Test images are from MeasTex texture database.

Image	$\sigma = 5$				$\sigma = 10$				$\sigma = 15$			
	σ_0	$\widehat{\sigma}$	Ours	Impv	σ_0	$\widehat{\sigma}$	Ours	Impv	σ_0	$\widehat{\sigma}$	Ours	Impv
Fabric	35.93	35.94	37.19	1.26	31.66	31.66	32.65	0.99	29.50	29.50	30.20	0.71
Food	35.35	35.37	36.06	0.71	30.53	30.54	31.16	0.63	28.08	28.08	28.53	0.45
Leaves	37.37	37.36	41.66	4.28	34.06	34.04	37.89	3.83	32.53	32.51	35.74	3.20

Table 2: Continued.

Image	$\sigma = 5$				$\sigma = 10$				$\sigma = 15$			
	σ_0	$\widehat{\sigma}$	Ours	Impv	σ_0	$\widehat{\sigma}$	Ours	Impv	σ_0	$\widehat{\sigma}$	Ours	Impv
Water	36.24	36.26	37.63	1.39	31.78	31.80	33.12	1.34	29.33	29.35	30.68	1.36
Wood	37.07	37.06	39.18	2.11	33.45	33.44	35.44	1.99	31.60	31.58	33.22	1.62
Image	$\sigma = 20$				$\sigma = 25$				$\sigma = 50$			
	σ_0	$\widehat{\sigma}$	Ours	Impv	σ_0	$\widehat{\sigma}$	Ours	Impv	σ_0	$\widehat{\sigma}$	Ours	Impv
Fabric	28.04	28.02	28.55	0.51	26.92	26.89	27.31	0.39	23.67	23.71	23.86	0.19
Food	26.48	26.47	26.80	0.33	25.30	25.28	25.56	0.25	22.01	22.02	22.31	0.30
Leaves	31.44	31.43	34.14	2.70	30.44	30.46	32.89	2.45	26.00	26.38	28.95	2.95
Water	27.60	27.57	29.02	1.42	26.24	26.11	27.75	1.51	22.15	22.09	23.96	1.80
Wood	30.26	30.23	31.62	1.36	29.16	29.11	30.44	1.28	25.33	25.40	26.72	1.39

Table 3: Comparison of several denoising results under different noise levels. Test images are from the Tid2008 database.

Image	$\sigma = 5$				$\sigma = 10$				$\sigma = 15$			
	σ_0	$\widehat{\sigma}$	Ours	Impv	σ_0	$\widehat{\sigma}$	Ours	Impv	σ_0	$\widehat{\sigma}$	Ours	Impv
Im01	35.13	35.10	35.37	0.24	30.40	30.42	30.85	0.46	28.05	28.06	28.44	0.39
Im02	36.41	36.45	37.24	0.84	32.61	32.63	33.64	1.03	30.84	30.86	31.86	1.02
Im03	36.75	36.78	39.16	2.41	33.04	33.05	35.34	2.30	31.33	31.33	33.28	1.95
Im04	36.82	36.89	38.16	1.34	33.31	33.33	34.90	1.58	31.73	31.74	33.29	1.55
Im05	35.60	35.60	36.14	0.53	30.84	30.85	31.41	0.57	28.29	28.31	28.84	0.55
Im06	35.40	35.37	36.06	0.66	30.58	30.59	31.57	0.99	28.17	28.18	29.18	1.01
Im07	36.67	36.69	38.48	1.81	32.80	32.80	34.23	1.43	30.88	30.87	32.01	1.12
Im08	35.35	35.21	35.55	0.19	30.64	30.60	31.16	0.51	28.18	28.16	28.76	0.58
Im09	36.50	36.55	37.81	1.31	32.55	32.56	33.87	1.32	30.61	30.60	31.75	1.14
Im10	36.55	36.61	37.71	1.16	32.64	32.65	33.92	1.28	30.72	30.71	31.87	1.14
Im11	35.72	35.72	36.45	0.73	31.28	31.29	32.15	0.87	29.08	29.08	29.87	0.79
Im12	36.40	36.42	37.66	1.26	32.57	32.58	34.05	1.48	30.81	30.80	32.15	1.34
Im13	34.47	34.20	34.44	−0.03	29.35	29.19	29.83	0.47	26.75	26.67	27.16	0.41
Im14	35.53	35.50	35.87	0.34	31.07	31.06	31.70	0.63	28.86	28.87	29.44	0.57
Im15	36.64	36.68	37.93	1.29	32.91	32.92	34.37	1.46	31.18	31.18	32.49	1.31
Im16	36.30	36.31	37.68	1.38	32.30	32.30	33.90	1.60	30.48	30.47	31.92	1.44
Im17	36.54	36.59	37.73	1.19	32.65	32.66	33.94	1.29	30.73	30.72	31.89	1.17
Im18	35.80	35.76	36.07	0.28	31.37	31.36	31.82	0.45	29.11	29.11	29.54	0.43
Im19	36.01	36.02	36.66	0.64	31.78	31.79	32.48	0.70	29.64	29.64	30.35	0.72
Im20	36.60	36.63	38.09	1.49	32.70	32.72	34.31	1.61	30.79	30.79	32.19	1.40
Im21	35.71	35.69	36.35	0.64	31.18	31.19	31.94	0.76	28.91	28.92	29.62	0.71
Im22	35.92	35.92	36.49	0.57	31.78	31.79	32.32	0.53	29.82	29.82	30.28	0.46
Im23	36.94	37.01	38.99	2.05	33.37	33.40	35.47	2.09	31.70	31.70	33.58	1.88
Im24	36.18	36.20	36.83	0.65	31.93	31.94	32.59	0.66	29.75	29.76	30.39	0.64
Im25	35.75	35.77	37.33	1.58	30.87	30.89	32.05	1.18	27.94	27.96	29.05	1.11
Image	$\sigma = 20$	$\sigma = 25$	$\sigma = 50$									
	σ_0	$\widehat{\sigma}$	Ours	Impv	σ_0	$\widehat{\sigma}$	Ours	Impv	σ_0	$\widehat{\sigma}$	Ours	Impv
Im01	26.54	26.56	26.93	0.38	25.49	25.52	25.92	0.43	22.85	22.87	23.25	0.40
Im02	29.72	29.74	30.72	1.01	28.87	28.90	29.92	1.05	26.34	26.36	27.78	1.44
Im03	30.22	30.23	31.93	1.71	29.34	29.37	30.98	1.64	26.28	26.45	28.44	2.15
Im04	30.73	30.73	32.24	1.50	29.96	29.95	31.46	1.50	27.45	27.43	28.94	1.49
Im05	26.59	26.63	27.14	0.55	25.33	25.38	25.87	0.54	21.85	21.89	22.26	0.41
Im06	26.66	26.69	27.58	0.92	25.62	25.66	26.40	0.78	23.00	23.03	23.37	0.37
Im07	29.60	29.58	30.50	0.91	28.59	28.58	29.42	0.83	25.39	25.40	26.28	0.89
Im08	26.55	26.55	27.17	0.63	25.36	25.38	25.99	0.63	21.99	22.00	22.60	0.62
Im09	29.34	29.33	30.33	0.99	28.39	28.38	29.29	0.90	25.51	25.50	26.35	0.84
Im10	29.46	29.45	30.50	1.04	28.52	28.51	29.45	0.93	25.72	25.72	26.47	0.75
Im11	27.65	27.66	28.35	0.71	26.57	26.61	27.23	0.66	23.44	23.52	24.25	0.82
Im12	29.71	29.70	30.89	1.18	28.91	28.90	29.99	1.08	26.48	26.46	27.54	1.05
Im13	25.09	25.08	25.64	0.54	23.95	23.96	24.48	0.53	21.10	21.12	21.20	0.10
Im14	27.46	27.47	27.93	0.47	26.44	26.45	26.86	0.41	23.64	23.64	23.95	0.32
Im15	30.06	30.06	31.25	1.20	29.16	29.19	30.37	1.21	26.02	26.16	27.83	1.80
Im16	29.38	29.37	30.61	1.24	28.60	28.60	29.71	1.11	26.43	26.43	27.49	1.05

Table 3: Continued.

Image	$\sigma = 5$				$\sigma = 10$				$\sigma = 15$			
	σ_0	$\widehat{\sigma}$	Ours	Impv	σ_0	$\widehat{\sigma}$	Ours	Impv	σ_0	$\widehat{\sigma}$	Ours	Impv
Im17	29.37	29.40	30.48	1.10	28.21	28.30	29.41	1.20	24.47	24.59	26.35	1.88
Im18	27.62	27.65	28.04	0.42	26.54	26.57	26.96	0.42	23.65	23.66	24.06	0.41
Im19	28.18	28.18	28.95	0.77	27.07	27.07	27.89	0.82	23.67	23.67	24.70	1.02
Im20	29.50	29.51	30.76	1.27	28.49	28.52	29.69	1.20	25.27	25.37	26.53	1.26
Im21	27.46	27.48	28.08	0.62	26.41	26.43	26.97	0.57	23.61	23.62	24.07	0.46
Im22	28.58	28.59	29.00	0.41	27.69	27.69	28.10	0.41	25.10	25.10	25.65	0.56
Im23	30.58	30.57	32.28	1.70	29.67	29.66	31.24	1.57	26.40	26.52	27.99	1.59
Im24	28.31	28.34	28.95	0.64	27.25	27.27	27.88	0.63	24.38	24.38	24.76	0.38
Im25	25.69	25.76	27.01	1.32	23.96	24.03	25.42	1.47	19.34	19.41	20.81	1.46

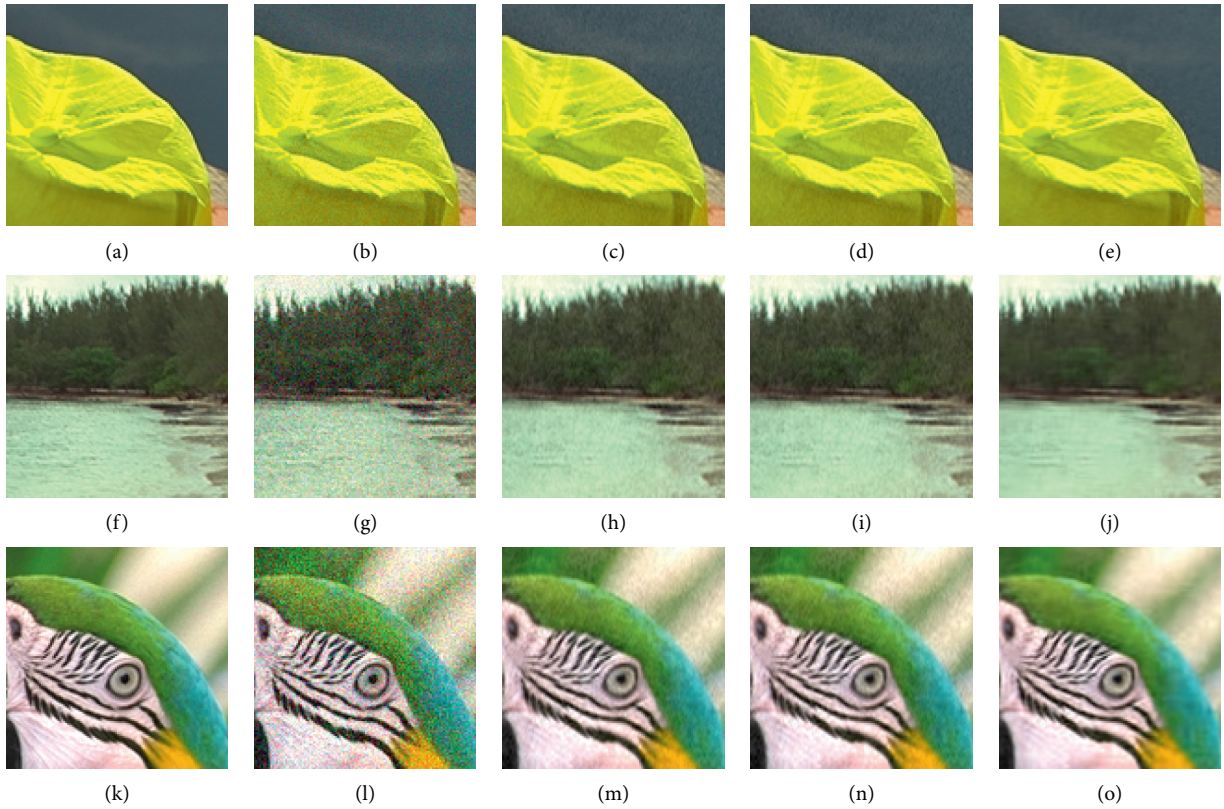

Figure 6: Visual and PSNR comparisons of several algorithms from color images. A cropped region measuring 120 × 120 in Im3, Im12, and Im23 images from the Tid2008 database is shown. The first column is a noise-free image, the second column is a noisy image corresponding noise level is below the image, the third and fourth columns are the results of curve denoising trained by using accurate and estimated noise levels, respectively, and the fifth column shows the denoising results of the proposed algorithm. (a) Im3. (b) $\sigma = 5$. (c) $\sigma_0 = 36.75$ (d) $\widehat{\sigma} = 36{:}78$. (e) ours = 39:16. (f) Im12. (g) $\sigma = 15$. (h) $\sigma_0 = 30.81$. (i) $\widehat{\sigma} = 30.80$. (j) ours = 32.15. (k) Im23. (l) $\sigma = 20$. (m) $\sigma_0 = 30.85$. (n) $\widehat{\sigma} = 30.57$. (o) ours = 32.28.

5. Conclusion and Future Work

A shrinkage curve learning denoising algorithm is an important kind of denoising algorithm, and an algorithm constructed by a shrinkage curve has typical nonlinear properties. However, the algorithm proposed in this paper can still boost shrinkage curve learning denoising. The proposed algorithm combines noise-level estimations and makes the denoising intensity of the denoiser match the noise image. Therefore, the denoising ability of the proposed algorithm is improved. Moreover, the algorithm proposed in this paper does not rely on external training samples, thus boosting the adaptability of the algorithm. Finally, the convergence of the proposed algorithm is discussed and the conditions for judging the convergence of the algorithm are obtained. Of course, a significant amount of time is taken in the proposed algorithm to train samples, mainly to solve a family of optimal shrinkage curves in the sense of least squares. If we do not pay attention to the analytical solutions of these curves, some of the most representative

computational intelligence algorithms can be used to solve the outstanding problems, including Harris hawks optimization [31], monarch butterfly optimization [32], and the earthworm optimization algorithm [33], among others. The moth search [34] algorithm can also solve outstanding problems and can possibly improve the efficiency of the algorithm. In addition, the establishment of a unified algorithm, so that the boosting algorithm can adapt to linear and nonlinear noise reduction, is also an approach that can be studied in the future. Finally, the denoiser $D_\sigma(\cdot)$ in the proposed algorithm should be associated with the noise level σ, which controls the denoising strength. How to improve the proposed algorithm if the denoiser is independent of σ commands requires further study.

Data Availability

The data and the MATLAB codes used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (Grant nos. 61763009, 61761030, and 62061016), the Doctoral Scientific Fund Project of Hubei Minzu University (Grant no. MD2020 B024), the High-level scientific research Achievement Cultivation project of Hubei Minzu University (Grant no.4205003), and the Nature Science Foundation of Enshi Polytechnic (Grant no. EZYQNZK201904). The authors would like to express our gratitude to the anonymous reviewers and editors for their valuable comments and suggestions, which led to the improvement of the original manuscript.

References

[1] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995.

[2] A. Buades, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 60–65, 2005.

[3] M. Aharon, M. Elad, and A. Bruckstein, "$rm K$-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[5] B. Goyal, A. Dogra, S. Agrawal, B. S. Sohi, and A. Sharma, "Image denoising review: from classical to state-of-the-art approaches," *Information Fusion*, vol. 55, pp. 220–244, 2020.

[6] A. Krull, T. O. Buchholz, and F. Jug, "Noise2void-learning denoising from single noisy im- ages," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2129–2137, California, CA, USA, 2019.

[7] S. Liu, M. Liu, P. Li, J. Zhao, Z. Zhu, and X. Wang, "SAR image denoising via sparse representation in shearlet domain based on continuous cycle spinning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 5, pp. 2985–2992, 2017.

[8] S. Liu, L. Gao, Y. Lei et al., "SAR speckle removal using hybrid frequency modulations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 3956–3966, 2021.

[9] Y. Romano and M. Elad, "Boosting of image denoising algorithms," *SIAM Journal on Imaging Sciences*, vol. 8, no. 2, pp. 1187–1219, 2015.

[10] J. W. Tuckey, *Exploratory Data Analysis*, Springer, New York, 1977.

[11] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 460–489, 2005.

[12] H. Talebi, X. Xiang Zhu, and P. Milanfar, "How to saif-ly boost denoising performance," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1470–1485, 2013.

[13] Z. Fang, X. Yi, and L. Tang, "An adaptive boosting algorithm for image denoising," *Mathematical Problems in Engineering*, vol. 2019, no. 3, 14 pages, Article ID 8365932, 2019.

[14] P. Milanfar, "A tour of modern image filtering: new insights and methods, both practical and theoretical," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 106–128, 2013.

[15] P. A. Knight, "The Sinkhorn-Knopp algorithm: convergence and applications," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 1, pp. 261–275, 2008.

[16] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *The Annals of Mathematical Statistics*, vol. 35, no. 2, pp. 876–879, 1964.

[17] F. M. Bayer, A. J. Kozakevicius, and R. J. Cintra, "An iterative wavelet threshold for signal denoising," *Signal Processing*, vol. 162, pp. 10–20, 2019.

[18] A. K. Bhandari, D. Kumar, A. Kumar, and G. K. Singh, "Optimal sub-band adaptive thresholding based edge preserved satellite image denoising using adaptive differential evolution algorithm," *Neurocomputing*, vol. 174, pp. 698–721, 2016.

[19] Y. Hel-Or and D. Shaked, "A discriminative approach for wavelet denoising," *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 443–457, 2008.

[20] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, "Deep learning on image denoising: an overview," *Neural Networks*, vol. 131, pp. 251–275, 2020.

[21] S. Lefkimmiatis, "Universal denoising networks: a novel CNN architecture for image denoising," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 3204–3213, June 2018.

[22] X. Jia, S. Liu, X. Feng, and Z. Lei, "FOCNet: A fractional optimal control network for image denoising," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, California, CA, USA, 2020.

[23] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer Science and Business Media, Germany, 2010.

[24] S. Pyatykh, J. Hesser, and L. Lei Zheng, "Image noise level estimation by principal component analysis," *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 687–699, 2013.

[25] X. Liu, M. Tanaka, and M. Okutomi, "Single-image noise level estimation for blind denoising," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5226–5237, 2013.
[26] P. Jiang and J.-Z. Zhang, "Fast and reliable noise level estimation based on local statistic," *Pattern Recognition Letters*, vol. 78, pp. 8–13, 2016.
[27] Z. Fang and X. Yi, "A novel natural image noise level estimation based on flat patches and local statistics," *Multimedia Tools and Applications*, vol. 78, no. 13, pp. 1–22, 2019.
[28] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play ADMM for image restoration: fixed-point convergence and applications," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 84–98, 2017.
[29] G. Smith and I. Burns, "Measuring texture classification algorithms," *Pattern Recognition Letters*, vol. 18, no. 14, pp. 1495–1501, 1997.
[30] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti, "Tid2008- a database for evaluation of full-reference visual quality assessment metrics," *Advances of Modern Radioelectronics*, vol. 10, no. 4, pp. 30–45, 2009.
[31] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.
[32] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing & Applications*, vol. 31, no. 7, pp. 1995–2014, 2019.
[33] G. G. Wang, S. Deb, and L. D. S. Coelho, "Earthworm optimisation algorithm: a bio-inspired metaheuristic algorithm for global optimisation problems," *International Journal of Bio-Inspired Computation*, vol. 12, no. 1, pp. 1–22, 2018.
[34] G. G. Wang, "Moth search algorithm: a bio-inspired metaheuristic algorithm for global opti- mization problems," *Memetic Computing*, vol. 10, no. 2, pp. 151–164, 2018.