Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 8365932, 14 pages
https://doi.org/10.1155/2019/8365932

Research Article

An Adaptive Boosting Algorithm for Image Denoising

Zhuang Fang,[1,2] **Xuming Yi**,[1] **and Liming Tang**[2]

[1]*School of Mathematics and Statistics, Wuhan University, Wuhan 430072, Hubei, China*
[2]*School of Science, Hubei Minzu University, Enshi 445000, Hubei, China*

Correspondence should be addressed to Xuming Yi; xmyi.math@whu.edu.cn

Received 4 July 2018; Revised 14 January 2019; Accepted 3 February 2019; Published 18 February 2019

Academic Editor: Ezequiel López-Rubio

Image denoising is an important problem in many fields of image processing. Boosting algorithm attracts extensive attention in recent years, which provides a general framework by strengthening the original noisy image. In such framework, many classical existing denoising algorithms can improve the denoising performance. However, the boosting step is fixed or nonadaptive; i.e., the noise level in iteration steps is set to be a constant. In this work, we propose a noise level estimation algorithm by combining the overestimation and underestimation results. Based on this, we further propose an adaptive boosting algorithm that excludes intricate parameter configuration. Moreover, we prove the convergence of the proposed algorithm. Experimental results that are obtained in this paper demonstrate the effectiveness of the proposed adaptive boosting algorithm. In addition, compared with the classical boosting algorithm, the proposed algorithm can get better performance in terms of visual quality and peak signal-to-noise ratio (PSNR).

1. Introduction

Image denoising is a fundamental problem in image processing, computer vision, pattern recognition, and so on. Consider a noisy image $\mathbf{y}$ modeled as

$$\mathbf{y} = \mathbf{x} + \epsilon, \tag{1}$$

where $\mathbf{x}$ is a clean image and ϵ is added white Gaussian noise with zero-mean and standard deviation σ. The goal of denoising is to resolve the clean image $\mathbf{x}$ in (1). To this end, many technique and methods, such as spatial adaptive filters, transform-domain methods, sparse representation, and processing of local patches have been explored to study this problem, which leads to state-of-the-art denoising algorithms (denoisers), including the NLM [1], K-SVD [2], EPLL [3], BM3D [4], BM3D-SAPCA [5], and LSSC [6].

Although the algorithms mentioned above are effective in denoising application, the performance can be improved by employing some specific techniques, such as "Twicing", Bregman iteration, and SAIF. In this paper, we focus on boosting skills originally used in machine learning, which improve denoising performance by multiple reusing [7–10] of weak denoisers to achieve strong ones. A general boosting algorithm [11] for image denoising can be expressed as

$$\widehat{\mathbf{x}}_{k+1} = \widehat{\mathbf{x}}_k + D_\sigma\left(\mathbf{y} - \widehat{\mathbf{x}}_k\right), \tag{2}$$

where $D_\sigma(\cdot)$ is a denoiser and $\widehat{\mathbf{x}}_k$ is the denoised image of the kth iteration. Essentially, the boosting algorithm (2) is to repeatedly denoise the residuals, $\mathbf{y} - \widehat{\mathbf{x}}_k$, and add them back to the denoised image. However, Talebi et al. [10] pointed out that the number of iterations must be tuned carefully since the sequence obtained by (2) is not always convergent to the best restoration.

To conquer this problem, Romano et al. [7] proposed another boosting algorithm, called SOS, which can be expressed as

$$\widehat{\mathbf{x}}_{k+1} = D_\sigma\left(\mathbf{y} + \widehat{\mathbf{x}}_k\right) - \widehat{\mathbf{x}}_k. \tag{3}$$

SOS strengthens the noisy image by adding portion of the denoised image $\widehat{\mathbf{x}}_k$ to the noisy image $\mathbf{y}$ before operating the next iteration denoiser and subtracting the same portion of the outcome. SOS has excellent performance in denoising application. In addition, it is guaranteed to converge to an optimal solution, which enables us to obtain stopping criteria easily.

We note that the denoisers $D_\sigma(\cdot)$ in (2) and (3) are invariant operators. In other words, the parameter σ in the denoisers is fixed regardless of the noise levels in the different iteration steps. The authors in [7] predefine a constant $\widetilde{\sigma}$ to estimate the initial noise level of $\mathbf{y}$ and then set $\sigma = t(\widetilde{\sigma})\widetilde{\sigma}$ with $t(\widetilde{\sigma}) > 1$ to estimate the noise level of $\mathbf{y} + \widehat{\mathbf{x}}_k$. In the following weaker denoising steps, the constant parameter σ is used in denoisers $D_\sigma(\cdot)$. Obviously, such scheme is not precise since $\widehat{\mathbf{x}}_k$s are the progressively restoring images which lead to a decreasing noise level estimation of $\mathbf{y} + \widehat{\mathbf{x}}_k$ with k increasing. To solve this problem, in this paper we propose an adaptive boosting algorithm for image denoising application. Adaptive denoisers $D_{\sigma_k}(\cdot)$ are used in boosting rather than an invariant operator $D_\sigma(\cdot)$, where the parameter σ_k is a noise level estimator of $\mathbf{y} + \widehat{\mathbf{x}}_k$.

The estimation of the noise level plays a key role in our adaptive boosting algorithm. In the last decades, many methods on noise level estimation have been proposed [12–20]. Among these methods, the blind evaluation of noise level in textured images was widely studied [15, 16], which leads to many noise level estimation methods that were wildly used in processing the highly textured images. Different from the methods that were reported in [15, 16], in this paper, we focus on patch-based methods which have received a lot of attention due to the sound theoretical basis and promise performance, such as PCA [17], WT [18], and others [19, 20]. Chen et al. [19] pointed out that PCA and WT methods could lead to an underestimation of the noise level by using the Blom's theorem [21]. Jiang et al. [20] proposed a noise level estimation method (called JZ in the following) based on the eigenvalues of covariance matrix of the flat patches. We in this paper prove that JZ method is an overestimation of the noise level. Combining WT underestimation and JZ overestimation, we propose a new estimator to obtain higher accurate estimation. The main contributions of this paper are as follows:

(i) We propose an adaptive boosting algorithm for image denoising application, in which an adaptive weaker denoiser $D_{\sigma_k}(\cdot)$ is introduced rather than an invariant operator $D_\sigma(\cdot)$ often used in the traditional boosting.

(ii) We prove that JZ method is an overestimation of the noise level by using Blom's theorem. In addition, combining WT underestimation and JZ overestimation, we propose a new estimator to obtain higher accurate estimation.

(iii) We prove the convergence of the proposed adaptive boosting algorithm. And some experiments are conducted to validate the proposed algorithm.

This paper is organized as follows: in Section 2, we present some related works concerning noise level estimation and image quality assessment. In Section 3, a noise level estimation algorithm and an adaptive boosting algorithm for image denoising application are proposed. In Section 4, the experiment procedure and the trial comparison between the proposed method and the initial denoising algorithm are described in detail. We summarize, conclude, and discuss the directions of future research in Section 5.

2. Previous Related Work

2.1. Eigenvalues and Noise Level. Liu et al. [18] gave a noise level estimation algorithm based on weak textured patches. According to their algorithm, the noise level of the image is estimated as follows:

$$\sigma_w^2 = \min_{1 \leqslant i \leqslant N} \left(e_w^i\right), \tag{4}$$

where e_w^i is the ith eigenvalues of $\Sigma_{\mathbf{P}_w}$, in which $\mathbf{P}_w$ is a set of weak textured patches that are selected by Algorithm 1 that was reported in [18], and $\Sigma_{\mathbf{P}_w}$ is the covariance matrix of $\mathbf{P}_w$. By selecting flat patches $\mathbf{P}_f$ in noisy image, Jiang et al. [20] gave another noise level estimation as follows:

$$\widehat{\sigma} = \frac{\min_{1 \leqslant i \leqslant N} \left(e_f^i\right)}{\sqrt{\rho}}, \tag{5}$$

where e_f^i denotes the ith eigenvalues of $\Sigma_{\mathbf{P}_f}$ and $\Sigma_{\mathbf{P}_f}$ is the covariance matrix of $\mathbf{P}_f$. In. (4) and (5), N is the element number of each patch. The basis of the above two methods is to extract a set of patches with Gaussian distribution. Chen et al. [19] proved the following theorems about the distribution of the eigenvalues.

Lemma 1 (see [19]). *Given a set of random variables* $\{\mathbf{y}_t[i]\}_{t=1}^s$, *with each element following Gaussian distribution* $N(0, \sigma^2)$ *independently, the distribution of the noise estimation* $\widehat{\sigma}_i^2 = (1/s)\sum_{t=1}^s \mathbf{y}_t[i]^2$ *converges to the distribution* $N(\sigma^2, 2\sigma^4/s)$ *when* s *becomes sufficiently large.*

Lemma 2 (see [19]). *Let* $\Phi(x) = (1/\sqrt{2\pi})\int_{-\infty}^{x} e^{-t^2/2}dt$ *denote the cumulative distribution function of a standard Gaussian distribution. Given that* N *independent random variables* $x_1, x_2, \ldots, x_N$ *generated from the normal distribution* $N(\sigma^2, \nu^2)$ *with order* $x_1 \geq x_2 \geq \cdots \geq x_N$, *then the expected value of* x_i *can be approximated by* $\mathbb{E}(x_i) \approx \sigma^2 + \Phi^{-1}((N - \alpha + 1 - i)/(N - 2\alpha + 1))\nu$ *with* $\alpha = 0.375$.

2.2. Natural Image Quality Evaluator (NIQE). The denoising algorithm is usually an iterative process in which the number of iterations needs to be selected such that the denoised image can achieve the best visual effect. For this problem, no-reference/blind image quality assessment models [22–26] are introduced. Recently, Mittal et al. [25] proposed a blind image quality assessment model called NIQE. In this method, the quality of a given image $\mathbf{y}$ is expressed as follows:

$$D_{niqe}(\mathbf{y}) = \sqrt{(\nu_1 - \nu_2)^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} (\nu_1 - \nu_2)}, \tag{6}$$

where ν_1, ν_2 and Σ_1, Σ_2 are the mean vectors and covariance matrices of the natural MVG model and the distorted image's MVG model. Compared with other methods, NIQE does not have to train on large databases of human opinions of distorted images and has low computation complexity. Thus, it is suited for determining the optimal numbers of iterations. More details about NIQE can be found in [25].

3. The Proposed Method

In this section, we discuss the theoretical foundation of the noise level estimation algorithm and give an estimation that linearly combines the overestimated and underestimated results to evaluate the noise levels. Then an adaptive boosting algorithm for image denoising is proposed. The details of these steps are presented in the following subsections.

3.1. Noise Level Estimation. Using Lemmas 1 and 2 we can establish the following result.

Corollary 3. *The estimation of σ_w^2 in (4) satisfies $\sigma_w^2 < \sigma^2$. Particularly, if $i_0 = N - \alpha + 1 - \Phi((\rho-1)\sqrt{s/2})(N - 2\alpha + 1)$, then $\sigma_f^2 > \sigma^2$, where*

$$\sigma_f^2 = \frac{\sigma^2 + \sqrt{2\sigma^4/s}\Phi^{-1}\left(\left(N - \alpha + 1 - \lfloor i_0 \rfloor\right)/\left(N - 2\alpha + 1\right)\right)}{\rho}, \tag{7}$$

Proof. If $i > (N+1)/2$, then $N + 1 - 2i < 0$. By adding $N - 2\alpha + 1$ to both sides, we have $2N - 2\alpha + 2 - 2i < N - 2\alpha + 1$; that is, $(N - \alpha + 1 - i)/(N - 2\alpha + 1) < 1/2$. Then, we have $\Phi^{-1}((N - \alpha + 1 - i)/(N - 2\alpha + 1)) < 0$. Thus $\sigma^2 + \Phi^{-1}((1-\alpha)/(N - 2\alpha + 1))\nu < \sigma^2$, with $\nu = \sqrt{2\sigma^4/s}$ and $i = N$. In (4), the last eigenvalue is selected as the noise level; it follows that $i = N$. From Lemma 2, the following relationship holds:

$$\sigma_w^2 = \sigma^2 + \Phi^{-1}\left(\frac{1-\alpha}{N - 2\alpha + 1}\right)\nu. \tag{8}$$

Thus $\sigma_w^2 < \sigma^2$.

Note that $(N+1)/2 \leqslant \lfloor i_0 \rfloor \leqslant N - \alpha + 1 - \Phi((\rho-1)\sqrt{s/2})(N - 2\alpha + 1)$. Then $N - \alpha + 1 - \lfloor i_0 \rfloor > \Phi((\rho-1)\sqrt{s/2})(N - 2\alpha + 1)$; that is, $(N - \alpha + 1 - \lfloor i_0 \rfloor)/(N - 2\alpha + 1) > \Phi((\rho-1)\sqrt{s/2})$. Because Φ is a monotonic function, we can obtain $\Phi^{-1}((N - \alpha + 1 - \lfloor i_0 \rfloor)/(N - 2\alpha + 1)) > (\rho-1)\sqrt{s/2}$. Then, $\Phi^{-1}((N - \alpha + 1 - \lfloor i_0 \rfloor)/(N - 2\alpha + 1))\sqrt{2/s} > \rho - 1$. Finally, we multiply both sides of the above equation by σ^2 and simplify it to obtain $\sigma_f^2 = (\sigma^2 + \sqrt{2/s}\sigma^2\Phi^{-1}((N - \alpha + 1 - \lfloor i_0 \rfloor)/(N - 2\alpha + 1)))/\rho > \sigma^2$. □

Theorem 4. *If $i_0 = N - \alpha + 1 - \Phi((\rho-1)\sqrt{s/2})(N - 2\alpha + 1)$, then noise level can be estimated by*

$$\sigma^2 = \frac{d_1\sigma_f^2 + d_2\sigma_w^2}{d_1 + d_2}, \tag{9}$$

where $d_1 = -\Phi^{-1}((1-\alpha)/(N - 2\alpha + 1))\sqrt{2/s}$ and $d_2 = ((1-\rho) + \Phi^{-1}((N - \alpha + 1 - [i_0])/(N - 2\alpha + 1))\sqrt{2/s})/\rho$.

Proof. From (7) and (8), we obtain the following:

$$\frac{\sigma^2 - \sigma_w^2}{\sigma_f^2 - \sigma^2} = \frac{-\Phi^{-1}\left(\left(1-\alpha\right)/\left(N - 2\alpha + 1\right)\right)\sqrt{2/s}\sigma^2}{\left(\left(\left(1-\rho\right) + \Phi^{-1}\left(\left(N - \alpha + 1 - \left[i_0\right]\right)/\left(N - 2\alpha + 1\right)\right)\sqrt{2/s}\right)/\rho\right)\sigma^2}. \tag{10}$$

Let $d_1 = -\Phi^{-1}((1-\alpha)/(N - 2\alpha + 1))\sqrt{2/s}$, $d_2 = ((1-\rho) + \Phi^{-1}((N - \alpha + 1 - [i_0])/(N - 2\alpha + 1))\sqrt{2/s})/\rho$. Then, (10) becomes

$$\frac{\sigma^2 - \sigma_w^2}{\sigma_f^2 - \sigma^2} = \frac{d_1}{d_2}. \tag{11}$$

Therefore, σ^2 can be solved by (11), that is,

$$\sigma^2 = \frac{d_1\sigma_f^2 + d_2\sigma_w^2}{d_1 + d_2}. \tag{12}$$

The collection of weak textured patches $\mathbf{P}_w$ and the flat patches $\mathbf{P}_f$ may not be the same. We extract the flat patches from the weak textured patches and denote it by $\mathbf{P}$. Then, both (7) and (8) hold on the collection $\mathbf{P}$, and the proposed algorithm for noise level estimation can be described in Algorithm 1. In experiment, we set $K = 3$ to ensure the efficiency of Algorithm 1 and obtain more accurate noise level. The results of the corresponding proof experiments can be seen in Section 4.1. In the following section, Algorithm 1 will be plugged in a new adaptive boosting algorithm. □

3.2. Adaptive Boosting Algorithm. Based on the analysis for denoiser in (3), the noise level is the main parameter in the iteration. We show that the denoiser can be improved by the following boosting procedure:

(1) Strengthen the signal by accumulating the previous denoised image to the noisy input image.

(2) Estimate the noise level σ_k of the strengthen image.

(3) Project the strengthen image to the range less than 1; i.e., divide the strengthen image by its infinite norm (maximum).

(4) Operate the denoiser $D_{c\sigma_k}$ on the project image and back-project the range of the outcome to match the clean image $\mathbf{x}$.

The main equations that describe the above procedure can be written as

$$\begin{aligned} \widehat{\mathbf{y}}_k &= \widehat{\mathbf{y}}_{k-1} + \lambda\widehat{\mathbf{x}}_{k-1}, \\ \widehat{\mathbf{x}}_k &= \frac{D_{c\sigma_k}\left(\widehat{\mathbf{y}}_k / \left\|\widehat{\mathbf{y}}_k\right\|_\infty\right)\left\|\widehat{\mathbf{y}}_k\right\|_\infty}{\left(1 + (k-1)\,\lambda\right)}, \end{aligned} \tag{13}$$

where the infinite norm $\|\cdot\|_\infty$ represents the maximum value of all elements, σ_k is the noise level of $\widehat{\mathbf{y}}_k$, the parameter c

Input: Noisy image $\mathbf{y}$.
Output Noise level $\widehat{\sigma}^2$.
Initialize Estimate the initial noise level σ_0, maximum number of iterations K.
While $k < K$**do**
1: $k = k + 1$;
2: Using Algorthm 1 in [18] to select weak textured patches $\mathbf{P}_w$.
3: For $\mathbf{P}_w$, use Algorithm 1 in [20] to select flat patches $\mathbf{P}$.
4: For all the patches in $\mathbf{P}$, calculate σ_f^2 using Eq. (7).
5: For all the patches in $\mathbf{P}$, calculate σ_w^2 using Eq. (8).
6: Calculate $\widehat{\sigma}_{k+1}^2$ using Eq. (9).
End while
Return Noise level $\widehat{\sigma}^2$.

Algorithm 1: Estimating the noise variance.

Input: Noisy image $\mathbf{y}$, denoising operator $D(\cdot)$.
Output: $\widehat{\mathbf{x}}_k$ An estimate for $\mathbf{x}$
Initialize: $\mathbf{y}_0 = \mathbf{y}$, $k = 0, c, \lambda$, $D_{niqe}(\widehat{\mathbf{x}}_0) = D_{niqe}(\mathbf{y})$, $\widehat{\mathbf{x}}_0 = 0$.
Main Iteration: Increment k by 1 and perform the following steps
1. $\widehat{\mathbf{y}}_k = \widehat{\mathbf{y}}_{k-1} + \lambda\widehat{\mathbf{x}}_{k-1}$.
2. σ_k = the noise level of $\widehat{\mathbf{y}}_k$ and it is estimated by Algorithm 1.
3. $\widehat{\mathbf{u}}_k = D_{c\sigma_k}(\widehat{\mathbf{y}}_k / \| \widehat{\mathbf{y}}_k\|_\infty)$.
4. $\widehat{\mathbf{x}}_k = (\widehat{\mathbf{u}}_k \| \widehat{\mathbf{y}}_k\|_\infty)/(1 + (k-1)\lambda)$.
Stopping Rule: If $D_{niqe}(\widehat{\mathbf{x}}_k) > D_{niqe}(\widehat{\mathbf{x}}_{k-1})$, stop. Otherwise, apply main iteration.
Return: $\widehat{\mathbf{x}}_k$ is obtained after k iterations.

Algorithm 2: Adaptive boosting algorithm for image denoising.

controls the strength of the denoiser, and λ controls the signal emphasis. Our full image denoising algorithm is given in Algorithm 2.

3.3. Convergence Analysis of Algorithm 2. In this section, bounded denoiser and linear denoiser are introduced and some proposition are analysed, then the convergence of Algorithm 2 is proved.

Definition 5 ((bounded denoiser) [27]). A bounded denoiser with a parameter σ is a function $D_\sigma(\mathbf{x}) : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ such that for any input $\mathbf{x} \in \mathbb{R}^n$,

$$\frac{\left\|D_\sigma(\mathbf{x}) - \mathbf{x}\right\|_2^2}{n} \leq G\sigma^2, \tag{14}$$

for some universal constant G independent of n and σ.

Definition 6 (linear denoiser). For the given constants σ, α, β, the denoiser $D_\sigma(\mathbf{x})$ is a linear operator, if

$$D_\sigma(\alpha\mathbf{x} + \beta\mathbf{y}) = \alpha D_\sigma(\mathbf{x}) + \beta D_\sigma(\mathbf{y}). \tag{15}$$

Milanfar [8] pointed out that many denoisers satisfy Definitions 5 and 6. Based on this, Romano et al. [7] proved the convergence of SOS algorithm by neglecting the nonlinear term. According to these definitions and the following properties, we prove the convergence of Algorithm 2.

Proposition 7. *For the given σ' and σ'', $\|D_{\sigma'}(\mathbf{x}) - D_{\sigma''}(\mathbf{x})\|_2 \leqslant \sqrt{nG}(\sigma' + \sigma'')$.*

Proof. Since $D_\sigma(\mathbf{x})$ is a bounded denoiser, we have $\|D_{\sigma'}(\mathbf{x}) - D_{\sigma''}(\mathbf{x})\|_2 = \|D_{\sigma'}(\mathbf{x}) - \mathbf{x} - D_{\sigma''}(\mathbf{x}) + \mathbf{x}\|_2 \leqslant \|D_{\sigma'}(\mathbf{x}) - \mathbf{x}\|_2 + \|D_{\sigma''}(\mathbf{x}) - \mathbf{x}\|_2 \leqslant \sqrt{nG}(\sigma' + \sigma'')$. □

In iterative equation (13), $\widehat{\mathbf{x}}_k$ is the kth iterative approximation of $\mathbf{x}$. Denote the error between $\widehat{\mathbf{x}}_k$ and $\mathbf{x}$ by ϵ_k; that is, $\widehat{\mathbf{x}}_k = \mathbf{x} + \epsilon_k$. Then it follows from the iterative equation $\widehat{\mathbf{y}}_{k+1} = \widehat{\mathbf{y}}_k + \lambda\widehat{\mathbf{x}}_k$ that

$$\widehat{\mathbf{y}}_k = \mathbf{y} + \lambda\left(\widehat{\mathbf{x}}_{k-1} + \cdots + \widehat{\mathbf{x}}_1\right). \tag{16}$$

Substituting $\mathbf{y} = \mathbf{x} + \epsilon$ and $\widehat{\mathbf{x}}_k = \mathbf{x} + \epsilon_k$ into (16), we can get

$$\widehat{\mathbf{y}}_k = [1 + \lambda(k-1)]\,\mathbf{x} + \left[\epsilon + \lambda\left(\epsilon_{k-1} + \cdots + \epsilon_1\right)\right]. \tag{17}$$

Denote $\widehat{\epsilon}_k = \epsilon + \lambda(\epsilon_{k-1} + \cdots + \epsilon_1)$. And let $\widehat{\epsilon} = \lim_{k\longrightarrow\infty}\widehat{\epsilon}_k$. Using the central limit theorem, $\widehat{\epsilon}$ is a normal random variable with $\mathbb{E}(\widehat{\epsilon}) = 0$. Furthermore, we denote the standard deviation of $\widehat{\epsilon}$ by $\overline{\sigma}$. This implies that $\widehat{\mathbf{y}}_k$ in (17) can be considered as clean portion $[1 + \lambda(k-1)]\mathbf{x}$ and a zero-mean Gaussian white noise with standard deviation $\overline{\sigma}$. Thus, the noise level of $\widehat{\mathbf{y}}_k$ estimated by Algorithm 1 is bounded. Then, we have the following proposition.

Proposition 8. *There exists a constant M satisfying $\|\sigma_k\| \leqslant M$.*

Proposition 9. *For all k, $\|\hat{\mathbf{x}}_k\|_\infty \leqslant \|\mathbf{y}\|_\infty$ and $\|\hat{\mathbf{y}}_k\|_\infty \leqslant (1 + k\lambda)\|\mathbf{y}\|_\infty$.*

Proof. Since $\hat{\mathbf{x}}_1 = D_{c\sigma_1}(\hat{\mathbf{y}}_1/\|\hat{\mathbf{y}}_1\|_\infty)\|\hat{\mathbf{y}}_1\|_\infty$, we have $\|\hat{\mathbf{x}}_1\|_\infty = \|D_{c\sigma_1}(\hat{\mathbf{y}}_1/\|\hat{\mathbf{y}}_1\|_\infty)\|\hat{\mathbf{y}}_1\|_\infty\|_\infty$, for $D_{c\sigma_k}(\hat{\mathbf{y}}_k/\|\hat{\mathbf{y}}_k\|_\infty) \in [0, 1]$. Then $\|\hat{\mathbf{x}}_1\|_\infty \leqslant \|\hat{\mathbf{y}}_1\|_\infty = \|\mathbf{y}\|_\infty$.

Assume $\|\hat{\mathbf{x}}_k\|_\infty \leqslant \|\mathbf{y}\|_\infty$. Then $\|\hat{\mathbf{x}}_{k+1}\|_\infty = \|(D_{c\sigma_{k+1}}(\hat{\mathbf{y}}_{k+1}/\|\hat{\mathbf{y}}_{k+1}\|_\infty)\|\hat{\mathbf{y}}_{k+1}\|_\infty)/(1 + k\lambda)\|_\infty \leqslant \|\hat{\mathbf{y}}_{k+1}\|_\infty/(1 + k\lambda)$. Furthermore, noting that $\hat{\mathbf{y}}_{k+1} = \hat{\mathbf{y}}_k + \lambda\hat{\mathbf{x}}_k$, we can get

$$\begin{aligned}\|\hat{\mathbf{x}}_{k+1}\|_\infty &\leqslant \left\|\frac{\mathbf{y} + \lambda(\hat{\mathbf{x}}_1 + \cdots + \hat{\mathbf{x}}_k)}{1 + k\lambda}\right\|_\infty \\ &\leqslant \frac{\|\mathbf{y}\|_\infty + \lambda(\|\hat{\mathbf{x}}_1\|_\infty + \cdots + \|\hat{\mathbf{x}}_k\|_\infty)}{1 + k\lambda} \\ &\leqslant \frac{\|\mathbf{y}\|_\infty + k\lambda\|\mathbf{y}\|_\infty}{1 + k\lambda} = \|\mathbf{y}\|_\infty.\end{aligned} \tag{18}$$

Therefore, the mathematical inductive method enables us to get $\|\hat{\mathbf{x}}_k\|_\infty \leqslant \|\boldsymbol{y}\|_\infty$.

Since $\hat{\mathbf{y}}_k = \hat{\mathbf{y}}_{k-1} + \lambda\hat{\mathbf{x}}_{k-1}$, we have $\|\hat{\mathbf{y}}_k\|_\infty = \|\hat{\mathbf{y}}_{k-1} + \lambda\hat{\mathbf{x}}_{k-1}\|_\infty \leqslant \|\hat{\mathbf{y}}_{k-1}\|_\infty + \lambda\|\hat{\mathbf{x}}_{k-1}\|_\infty \leqslant \|\hat{\mathbf{y}}_{k-2}\|_\infty + \lambda\|\hat{\mathbf{x}}_{k-2}\|_\infty + \lambda\|\mathbf{y}\|_\infty \leqslant \cdots \leqslant (1 + k\lambda)\|\mathbf{y}\|_\infty$. □

Proposition 10. *For any positive integer $k \in Z^+$, the following inequalities hold:*

$$\begin{aligned}&\|\hat{\mathbf{x}}_k\|_2 \leqslant \frac{n(1 + k\lambda)}{1 + (k-1)\lambda}\|\mathbf{y}\|_\infty, \\ &\text{and } \|\hat{\mathbf{y}}_k\|_2 \leqslant \|\mathbf{y}\|_2 + n\lambda(1 + \lambda)(k-1)\|\mathbf{y}\|_\infty.\end{aligned} \tag{19}$$

Proof. For any given $\hat{\mathbf{y}}_k \in \mathbb{R}^n$, $\hat{\mathbf{y}}_k = (\hat{y}_k^1, \hat{y}_k^2, \ldots, \hat{y}_k^n)$, it holds that $\hat{y}_k^i/\|\hat{\mathbf{y}}_k\|_\infty \in [0, 1]$. Therefore, for all noise level σ_k, we have

$$\left\|D_{c\sigma_k}\left(\frac{\hat{\mathbf{y}}_k}{\|\hat{\mathbf{y}}_k\|_\infty}\right)\right\|_2 \leqslant n. \tag{20}$$

Noting that

$$\begin{aligned}\|\hat{\mathbf{x}}_k\|_2 &= \left\|\frac{D_{c\sigma_k}(\hat{\mathbf{y}}_k/\|\hat{\mathbf{y}}_k\|_\infty)\|\hat{\mathbf{y}}_k\|_\infty}{1 + (k-1)\lambda}\right\|_2 \\ &= \frac{1}{1 + (k-1)\lambda}\left\|D_{c\sigma_k}\left(\frac{\hat{\mathbf{y}}_k}{\|\hat{\mathbf{y}}_k\|_\infty}\right)\right\|_2\|\hat{\mathbf{y}}_k\|_\infty,\end{aligned} \tag{21}$$

and employing (20) and Proposition 9, we have

$$\|\hat{\mathbf{x}}_k\|_2 \leqslant \frac{n(1 + k\lambda)}{1 + (k-1)\lambda}\|\mathbf{y}\|_\infty. \tag{22}$$

Using the iteration $\hat{\mathbf{y}}_{k+1} = \hat{\mathbf{y}}_k + \lambda\hat{\mathbf{x}}_k$, we have $\hat{\mathbf{y}}_k = \hat{\mathbf{y}} + \lambda(\hat{\mathbf{x}}_{k-1} + \cdots + \hat{\mathbf{x}}_1)$. Then

$$\|\hat{\mathbf{y}}_k\|_2 \leqslant \|\mathbf{y}\|_2 + \lambda(\|\hat{\mathbf{x}}_{k-1}\|_2 + \cdots + \|\hat{\mathbf{x}}_1\|_2). \tag{23}$$

Substituting (22) into (23), we get

$$\begin{aligned}\|\hat{\mathbf{y}}_k\|_2 \leqslant \|\mathbf{y}\|_2 + \lambda\Bigg\{&\left(\frac{n[1 + (k-1)\lambda]\|\mathbf{y}\|_\infty}{1 + (k-2)\lambda}\right. \\ &\left. + \frac{n[1 + (k-2)\lambda]\|\mathbf{y}\|_\infty}{1 + (k-3)\lambda} + \cdots + n(1 + \lambda)\|\mathbf{y}\|_\infty\right\}.\end{aligned} \tag{24}$$

Since $(1 + (k-1)\lambda)/(1 + (k-2)\lambda) < 1 - \lambda$, then we have

$$\|\hat{\mathbf{y}}_k\|_2 \leqslant \|\mathbf{y}\|_2 + n\lambda(1 + \lambda)(k-1)\|\mathbf{y}\|_\infty. \tag{25}$$

This completes the proof. □

Next, we further consider the convergence of Algorithm 2.

Theorem 11. *If $D_\sigma(\cdot)$ is bounded and linear denoiser, then the main iterations of Algorithm 2 are convergent; that is, $\|\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k\|_2 \longrightarrow 0$ as $k \longrightarrow \infty$.*

Proof. It is easy to obtain

$$\begin{aligned}\left\|D_{c\sigma_k}(\hat{\mathbf{x}}_k)\right\|_2 &= \left\|D_{c\sigma_k}(\hat{\mathbf{x}}_k) - \hat{\mathbf{x}}_k + \hat{\mathbf{x}}_k\right\|_2 \\ &\leqslant \left\|D_{c\sigma_k}(\hat{\mathbf{x}}_k) - \hat{\mathbf{x}}_k\right\|_2 + \|\hat{\mathbf{x}}_k\|_2.\end{aligned} \tag{26}$$

Further applying Definition 5 and substituting (22) into (26), we have

$$\left\|D_{c\sigma_k}(\hat{\mathbf{x}}_k)\right\|_2 \leqslant \sqrt{nG}c\sigma_k + \frac{n(1 + k\lambda)}{1 + (k-1)\lambda}\|\mathbf{y}\|_\infty. \tag{27}$$

Similarly,

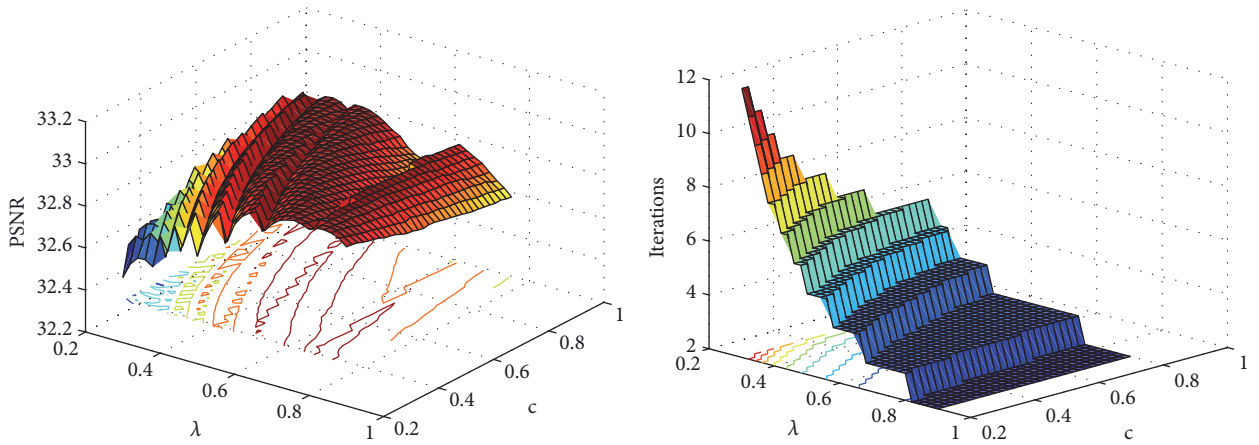
$$\begin{aligned}\left\|D_{c\sigma_k}(\hat{\mathbf{y}}_k)\right\|_2 &= \left\|D_{c\sigma_k}(\hat{\mathbf{y}}_k) - \hat{\mathbf{y}}_k + \hat{\mathbf{y}}_k\right\|_2 \\ &\leqslant \left\|D_{c\sigma_k}(\hat{\mathbf{y}}_k) - \hat{\mathbf{y}}_k\right\|_2 + \|\hat{\mathbf{y}}_k\|_2.\end{aligned} \tag{28}$$

Applying Definition 5 and substituting (25) into (28), we obtain

$$\begin{aligned}\left\|D_{c\sigma_k}(\hat{\mathbf{y}}_k)\right\|_2 \leqslant{}& \sqrt{nG}c\sigma_k + \|\mathbf{y}\|_2 \\ &+ n\lambda(1 + \lambda)(k-1)\|\mathbf{y}\|_\infty.\end{aligned} \tag{29}$$

On the other hand, the iteration $\hat{\mathbf{y}}_{k+1} = \hat{\mathbf{y}}_k + \lambda\hat{\mathbf{x}}_k$ leads to

$$\begin{aligned}&\left\|\frac{1}{1 + k\lambda}D_{c\sigma_k}(\hat{\mathbf{y}}_{k+1}) - \frac{1}{1 + (k-1)\lambda}D_{c\sigma_k}(\hat{\mathbf{y}}_k)\right\|_2 \\ &= \left\|\frac{1}{1 + k\lambda}D_{c\sigma_k}(\hat{\mathbf{y}}_k + \lambda\hat{\mathbf{x}}_k)\right. \\ &\left. - \frac{1}{1 + (k-1)\lambda}D_{c\sigma_k}(\hat{\mathbf{y}}_k)\right\|_2 = \left\|\frac{1}{1 + k\lambda}D_{c\sigma_k}(\hat{\mathbf{y}}_k)\right. \\ &\left. + \frac{\lambda}{1 + k\lambda}D_{c\sigma_k}(\hat{\mathbf{x}}_k) - \frac{1}{1 + (k-1)\lambda}D_{c\sigma_k}(\hat{\mathbf{y}}_k)\right\|_2 \\ &\leqslant \left\|\frac{1}{1 + k\lambda}D_{c\sigma_k}(\hat{\mathbf{y}}_k) - \frac{1}{1 + (k-1)\lambda}D_{c\sigma_k}(\hat{\mathbf{y}}_k)\right\|_2 \\ &+ \frac{\lambda}{1 + k\lambda}\left\|D_{c\sigma_k}(\hat{\mathbf{x}}_k)\right\|_2 \\ &= \frac{\lambda}{(1 + k\lambda)[1 + (k-1)\lambda]}\left\|D_{c\sigma_k}(\hat{\mathbf{y}}_k)\right\|_2 \\ &+ \frac{\lambda}{1 + k\lambda}\left\|D_{c\sigma_k}(\hat{\mathbf{x}}_k)\right\|_2.\end{aligned} \tag{30}$$



(a) Relationship among λ, c, and PSNR

(b) Relationship among λ, c, and iterations

FIGURE 1: Demonstration of the effect of λ and c on the proposed algorithm outcome for the BM3D ($\sigma = 20$).

Substituting (27) and (29) into (30), we have

$$\begin{aligned}&\left\|\frac{1}{1+k\lambda}D_{c\sigma_k}(\widehat{\mathbf{y}}_{k+1})-\frac{1}{1+(k-1)\lambda}D_{c\sigma_k}(\widehat{\mathbf{y}}_k)\right\|_2\\&\leqslant\frac{\lambda\left(\sqrt{nG}c\sigma_k+\|\mathbf{y}\|_2+n\lambda(1+\lambda)(k-1)\|\mathbf{y}\|_\infty\right)}{(1+k\lambda)[1+(k-1)\lambda]}\\&+\frac{\lambda\left(\sqrt{nG}c\sigma_k+(n(1+k\lambda)/(1+(k-1)\lambda))\|\mathbf{y}\|_\infty\right)}{1+k\lambda}.\end{aligned}\tag{31}$$

Finally,

$$\begin{aligned}\|\widehat{\mathbf{x}}_{k+1}-\widehat{\mathbf{x}}_k\|_2&=\left\|\frac{1}{1+k\lambda}D_{c\sigma_{k+1}}(\widehat{\mathbf{y}}_{k+1})\right.\\&\left.-\frac{1}{1+(k-1)\lambda}D_{c\sigma_k}(\widehat{\mathbf{y}}_k)\right\|_2=\left\|\frac{1}{1+k\lambda}D_{c\sigma_{k+1}}(\widehat{\mathbf{y}}_{k+1})\right.\\&\left.-\frac{1}{1+k\lambda}D_{c\sigma_k}(\widehat{\mathbf{y}}_{k+1})+\frac{1}{1+k\lambda}D_{c\sigma_k}(\widehat{\mathbf{y}}_{k+1})\right.\\&\left.-\frac{1}{1+(k-1)\lambda}D_{c\sigma_k}(\widehat{\mathbf{y}}_k)\right\|_2\\&\leqslant\frac{1}{1+k\lambda}\left\|D_{c\sigma_{k+1}}(\widehat{\mathbf{y}}_{k+1})-D_{c\sigma_k}(\widehat{\mathbf{y}}_{k+1})\right\|_2\\&+\left\|\frac{D_{c\sigma_k}(\widehat{\mathbf{y}}_{k+1})}{1+k\lambda}-\frac{D_{c\sigma_k}(\widehat{\mathbf{y}}_k)}{1+(k-1)\lambda}\right\|_2.\end{aligned}\tag{32}$$

Applying Proposition 7 and substituting (31) into (32), we obtain

$$\begin{aligned}\|\widehat{\mathbf{x}}_{k+1}-\widehat{\mathbf{x}}_k\|_2&\leqslant\frac{\sqrt{nG}c(\sigma_{k+1}+\sigma_k)}{1+k\lambda}\\&+\frac{\lambda\left(\sqrt{nG}c\sigma_k+(n(1+k\lambda)/(1+(k-1)\lambda))\|\mathbf{y}\|_\infty\right)}{1+k\lambda}\\&+\frac{\lambda\left(\sqrt{nG}c\sigma_k+\|\mathbf{y}\|_2+n\lambda(1+\lambda)(k-1)\|\mathbf{y}\|_\infty\right)}{(1+k\lambda)[1+(k-1)\lambda]}.\end{aligned}\tag{33}$$

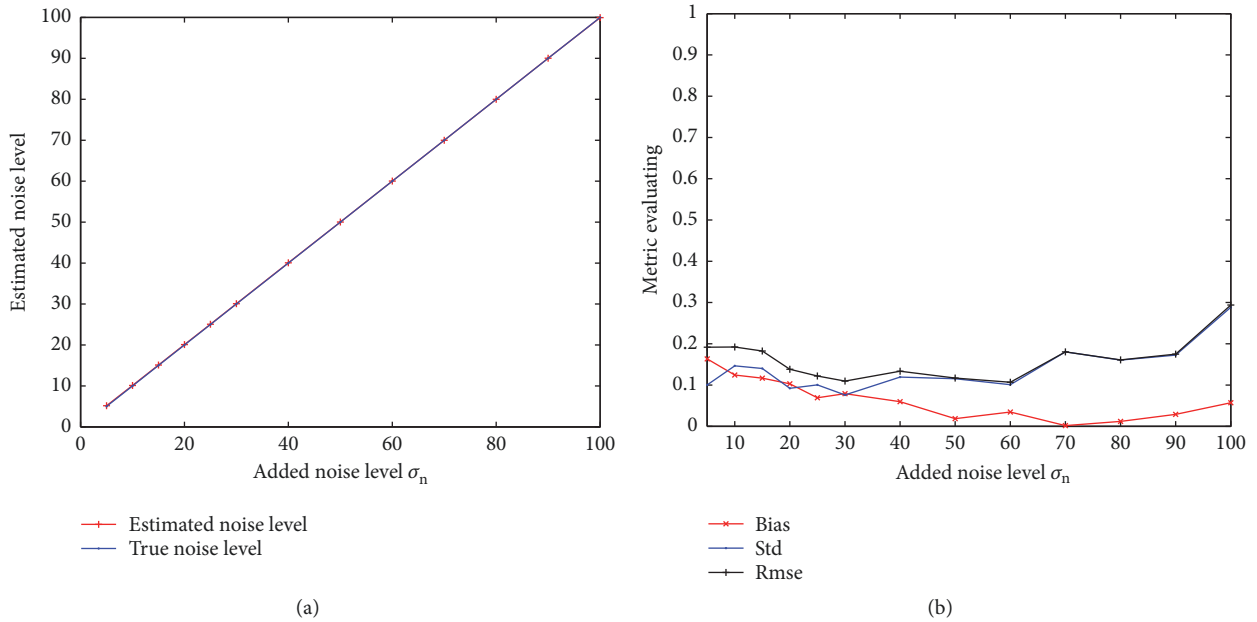
From Proposition 8 both σ_{k+1} and σ_k are all bounded. And since $\|\mathbf{y}\|_2$ and $\|\mathbf{y}\|_\infty$ are finite, we can get $\|\widehat{\mathbf{x}}_{k+1}-\widehat{\mathbf{x}}_k\|_2\longrightarrow 0$ as $k\longrightarrow\infty$, by taking the limit on both sides of (33). □

3.4. Parameters Configuration. In Algorithm 2, we select the state-of-the-art algorithms: BM3D and BM3D-SAPCA. The source code of these algorithms can be obtained from the original authors and we use the default parameters. When the denoiser is given, the parameters in Algorithm 2 are λ and c. In the remainder of this section, we mainly discuss the influence of λ and c on the final results.

Considering $\lambda\in[0.3,0.9]$ and $c\in[0.2,0.8]$ and discreting $[0.3,0.9]\times[0.2,0.8]$ by using a step-size 0.1×0.1, we get the parameters set $\{(\lambda_i,c_j)\}$. Then, we introduce these parameters into the proposed Algorithm 2 and apply the proposed algorithm to noisy image. Figure 1(a) shows the relationship between (λ,c) and PSNR defined as $20\log_{10}(255/\sqrt{MSE})$, where MSE is the mean squared error between the original image and its denoised version. The relationship between (λ,c) and optimizing truncation iteration number is shown in Figure 1(b). It can be seen from Figure 1, when the parameters λ and c are small, we can get the smallest PSNR and the largest iteration number. In this case, the efficiency of the denoising is the lowest. When both λ and c achieve the highest value, the number of the iteration is the minimum, but the PSNR is not the maximum value. It is obvious that the PSNR has some volatility with λ. The iteration number of the $\lambda-c$ area corresponding to the optimal PSNR value maintains consistency. In order to get the best denoising performance and reduce the computation complexity, suitable parameters λ and c must be determined. According to the results that are present in Figure 1, the PSNR is highest and the number of iterations is acceptable when $\lambda=0.635$ and $c=0.472$. Therefore, we select (0.635, 0.472) in the following experiment.

(a) Foreman (b) Lena (c) House (d) Fingerprint (e) Peppers

Figure 2: Visualization of the test images.



(a) (b)

Figure 3: Noise level estimation on TID2008. (a) Noise level estimation results. (b) The evaluation performance of the noise level estimation, accuracy, reliability, and overall performance.

4. Results

4.1. Noise Level Estimation Results. Usually, the value of Bias, Std, and RMSE are used to evaluate the noise estimation performance. We use well-known criteria: accuracy, reliability, and overall performance which have been considered in most literatures. In detail,

$$Bias\left(\widehat{\sigma}\right) = \mathbb{E}\left|\sigma - \mathbb{E}\left(\widehat{\sigma}\right)\right|, \tag{34}$$

$$Std\left(\widehat{\sigma}\right) = \sqrt{\mathbb{E}\left[\sigma - \mathbb{E}\left(\widehat{\sigma}\right)\right]^2}, \tag{35}$$

and

$$RMSE\left(\widehat{\sigma}\right) = \sqrt{Bias^2\left(\widehat{\sigma}\right) + Std^2\left(\widehat{\sigma}\right)}. \tag{36}$$

We test our method on Tampere image datasets (TID2008) [28] which contains 25 images of size 512×384. All images in this datasets are disturbed by zero-mean Gaussian noise with different standard deviations $\sigma = 5$, 10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, and 100. According to Figure 3(a), the estimated results are almost as same as the true noise level. As shown in Figure 3(b), the value of Bias, Std, and RMSE are all close to zero, which illustrates the accuracy of the Algorithm 1.

4.2. Denoising Results

4.2.1. Objective Measurement Results. In this section, we provide detail results of the proposed methods. Based on the completely blind image quality assessment method NIQE, we always consider the following stopping rule: $D_{niqe}(\widehat{\mathbf{x}}_i) < D_{niqe}(\widehat{\mathbf{x}}_{i-1}), i = 1, 2, \ldots, k-1$, and $D_{niqe}(\widehat{\mathbf{x}}_k) > D_{niqe}(\widehat{\mathbf{x}}_{k-1})$. The final outcome of Algorithm 2 is $\widehat{\mathbf{x}}_k$. We evaluate the competing methods on standard image processing test dataset (SIPTD) including Forman, Lena, House, Fingerprint, and Peppers, whose scenes are shown in Figure 2. These images are extensively tested in image processing. Thus it is convenient to compare our method with other methods under the same condition. All the test images are corrupted by an additive zero-mean Gaussian noise with a variance σ^2. The denoising performance is evaluated by using the PSNR, structural similarity (SSIM) [29], and dissimilar index (DSI) [30]. All

Table 1: Comparison among the denoising results (PSNR, SSIM, and DSI) of BM3D and the proposed algorithm.

BM3D($\lambda = 0.635$, $c = 0.472$)									
σ	Forman			Lena			House		
	σ_0	SOS	OURS	σ_0	SOS	OURS	σ_0	SOS	OURS
10	37.24	37.24	37.24	35.87	35.87	35.87	36.65	36.65	36.65
	0.9375	0.9368	0.9368	0.9149	**0.9141**	0.9154	0.9187	**0.9163**	0.9220
	2.1859	2.3486	**1.7016**	3.1644	3.3521	**2.8190**	2.7045	2.8983	**2.0538**
20	34.43	34.43	**34.50**	32.98	32.99	**33.01**	33.67	**33.69**	33.67
	0.9061	**0.9085**	0.9067	0.8764	0.8751	**0.8771**	**0.8684**	0.8657	0.8663
	3.3650	3.7503	3.6073	**4.7598**	5.2280	5.0152	**4.4826**	4.9408	5.0619
25	33.44	33.46	**33.60**	32.02	32.04	**32.07**	32.79	32.80	**32.85**
	0.8922	0.8978	**0.8983**	0.8599	0.8591	**0.8619**	**0.8566**	0.8555	0.8558
	3.6520	4.2599	4.3283	**5.4374**	6.1221	6.2237	**4.9677**	5.5707	5.9420
50	30.04	30.13	**30.18**	28.78	28.81	**29.04**	29.38	**29.42**	29.79
	0.8310	0.8453	**0.8454**	0.7860	0.7926	**0.8014**	0.7998	0.8057	**0.8159**
	4.7899	5.3656	6.6166	**10.191**	10.757	11.867	**7.7156**	8.3724	9.3492
σ	Fingerprint			Pepper			Average		
	σ_0	SOS	OURS	σ_0	SOS	OURS	σ_0	SOS	OURS
10	32.46	32.46	**32.51**	34.58	**34.64**	34.59	35.36	**35.37**	**35.37**
	0.9690	**0.9690**	0.9682	**0.9274**	0.9267	0.9264	0.9336	0.9326	**0.9338**
	0.7034	0.7205	**0.4263**	2.5305	2.6643	**2.0317**	2.2577	2.3968	**1.8065**
20	28.81	28.81	**28.82**	31.23	**31.27**	**31.27**	32.22	32.24	**32.25**
	0.9305	0.9281	**0.9308**	0.8849	0.8855	**0.8859**	0.8933	0.8926	**0.8934**
	6.2994	6.4851	**5.3902**	5.7494	6.1046	**4.8925**	4.9312	5.3018	**4.7934**
25	**27.70**	**27.70**	27.69	30.15	30.16	**30.19**	31.22	31.23	**31.28**
	0.9121	0.9072	**0.9122**	0.8667	0.8678	**0.8681**	0.8775	0.8775	**0.8793**
	10.310	10.657	**10.260**	7.6232	8.1311	**7.0077**	**6.3980**	6.9482	6.7523
50	24.34	24.36	**24.53**	26.32	26.38	**26.83**	**27.77**	27.82	28.07
	0.8253	**0.8143**	0.8367	**0.7700**	0.7797	**0.7952**	0.8024	0.8075	**0.8189**
	35.098	34.955	**27.840**	**18.296**	18.521	18.802	15.218	15.594	**14.895**

results obtained by competing denoising method are shown in Tables 1 and 2. The results that appear in the σ_0 column are obtained by applying the BM3D or BM3D-SAPCA on $\mathbf{y}$ using the accurate noise standard deviation. The results that appear in the SOS and "OURS" columns are obtained by applying SOS that was reported [7] and the proposed Algorithm 2, respectively. The best results for each image and noise level are highlighted. In contrast, Algorithm 2 achieves higher PSNR and SSIM than those of the other schemes. Compared with BM3D, PSNR obtained by Algorithm 2 achieves about 0.45dB improvement on the image Peppers with the noise level 50. The results presented in Table 3 indicate that the proposed approach achieves the highest average improvement of the PSNR and SSIM among all compared methods. Hence, for SIPTD dataset, the proposed algorithm is effective.

Moreover, we perform Algorithm 2 on all the grass images from the MeasTex texture dataset [31] which contain rich texture features. The comparison results are presented in Table 4. Compared with SOS methods, the average improvements of PSNR and SSIM demonstrate significant comparative advantages for almost all noise levels. It is clear that the proposed algorithm offers better restoration of texture images.

Finally, we use DSI (the Matlab realization of the DSI metric are available at http://ponomarenko.info/flt.htm.) which obtains the largest spearman rank order correlation coefficient values to mean opinion scores to evaluate the visual quality of the denoised images. According to the results present in Tables 3 and 4, the proposed Algorithm 2 obtained lower average DSI value for each denoiser on the considering datasets in most case.

4.2.2. Detail Contrast. In this section, we compare the performance of considered denoising algorithms in preserving image details. Figure 5 shows the fragments of noisy ($\sigma = 25$) Baboon image and the corresponding images denoised by different denoisers. The first row is the original image and its noisy version. The second row shows the denoised result corresponding to different parameters: exact σ, SOS, and *ours* by using the denoiser BM3D. The third row is the denoised result obtained by the denoiser BM3D-SAPCA. The enlarge fragments in each subfigure are helpful for demonstrating the good quality of the denoised images of faithful detail preservation. Figures 5(f) and 5(i) show more details close to the original image.

Table 2: Comparison among the denoising results (PSNR, SSIM, and DSI) of BM3D-SAPCA and the proposed algorithm.

BM3D-SAPCA($\lambda = 0.635$; $c = 0.472$)									
σ		Forman			Lena			House	
	σ_0	SOS	OURS	σ_0	SOS	OURS	σ_0	SOS	OURS
10	37.52	37.54	**37.54**	36.02	**36.05**	36.02	**37.01**	36.95	**37.01**
	0.9396	0.9397	**0.9398**	**0.9168**	**0.9168**	0.9166	0.9274	0.9268	**0.9279**
	1.8436	2.0052	**1.5171**	**3.0206**	3.2788	3.2277	2.2578	2.4728	**2.2514**
20	34.62	34.62	**34.68**	33.19	**33.23**	33.19	33.90	33.91	**33.92**
	0.9054	**0.9060**	0.9055	0.8796	0.8804	**0.8808**	0.8727	0.8719	**0.8735**
	3.4477	3.8614	3.8084	**4.9624**	5.5129	5.5065	**4.6415**	5.0781	5.0674
25	33.69	33.77	**33.81**	32.22	32.22	**32.23**	32.96	32.92	**33.02**
	0.8917	0.8935	**0.8935**	0.8644	0.8656	**0.8659**	**0.8588**	0.8581	0.8585
	3.8153	4.3223	4.5121	**5.9266**	6.5503	6.8346	**5.4209**	6.0023	6.1307
50	30.32	30.50	**30.71**	29.05	29.09	**29.26**	29.53	29.63	**29.88**
	0.8402	0.8430	**0.8497**	0.8022	0.8053	**0.8080**	0.8045	0.8095	**0.8162**
	5.2345	5.7885	7.3369	**9.6687**	10.7764	13.6531	**7.5772**	7.9921	9.72155
σ		Fingerprint			Peppers			Average	
	σ_0	SOS	OURS	σ_0	SOS	OURS	σ_0	SOS	OURS
10	32.64	32.66	**32.69**	34.94	34.95	**34.96**	35.63	35.63	**35.64**
	0.9703	0.9704	**0.9706**	0.9284	**0.9284**	0.9280	0.9365	0.9364	**0.9366**
	0.7503	0.7795	0.8502	**2.2488**	2.4581	3.2900	**2.0242**	2.1989	2.2273
20	28.94	28.96	**29.00**	31.55	31.56	**31.59**	32.44	32.46	**32.48**
	0.9328	0.9329	**0.9329**	0.8868	0.8875	**0.8884**	0.8955	0.8957	**0.8962**
	6.8078	7.1170	7.4005	**5.3362**	5.9814	5.9419	**5.0391**	5.5102	5.5449
25	27.81	27.83	**27.86**	30.43	30.44	**30.49**	31.42	31.44	**31.48**
	0.9145	**0.9146**	0.9135	0.8692	0.8699	**0.8706**	0.8797	0.8803	**0.8804**
	11.629	12.163	13.8800	**7.2262**	7.8891	8.0591	**6.8035**	7.3855	7.8833
50	24.53	**24.55**	24.48	27.00	27.05	**27.06**	28.09	28.16	**28.28**
	0.8354	**0.8360**	0.8286	**0.7945**	0.7982	**0.8005**	0.8154	0.8184	**0.8206**
	36.143	39.1979	52.0200	**16.5315**	17.2333	20.4961	**15.031**	16.198	20.646

Table 3: Comparison of average improvement between SOS and Algorithm 2 on SIPTD dataset.

	BM3D				BM3D-SAPCA			
	PSNR				PSNR			
σ	10	20	25	50	10	20	25	50
SOS	0.01	0.01	0.01	0.05	0.01	0.02	**0.02**	**0.07**
Ours	**0.01**	**0.03**	**0.06**	**0.30**	**0.27**	**0.22**	0.01	0.04
	SSIM				SSIM			
SOS	0.0005	0.0002	0.0009	0.0051	0.0001	0.0003	**0.0006**	**0.0031**
Ours	**0.0007**	**0.0003**	**0.0018**	**0.0165**	**0.0202**	**0.0193**	0.0001	0.0008
	DSI				DSI			
SOS	0.1390	0.3705	0.5502	0.3759	**0.1747**	**0.4710**	**0.5819**	**1.1667**
Ours	**-0.4513**	**-0.1379**	**0.3543**	**-0.3234**	0.2031	0.5058	1.0797	5.6146

Figure 6 shows the fragments of noisy ($\sigma = 20$) fingerprint image and the corresponding images denoised by different denoisers. According to Figure 6, our approach outperforms BM3D, SOS, and BM3D-SAPCA, numerically and visually.

Fragments of noisy ($\sigma = 50$) and denoised Foreman are shown in Figure 7. For this relatively high level of noise, the proposed Algorithm 2 attains a good preservation of sharp details, such as the lines on the wall in Foreman image. Meanwhile the smooth regions, such as the hat in Foreman image, are also of a good preservation. All the denoised images obtained by using the proposed Algorithm 2 have the fewest disturbing artifacts.

4.2.3. Computational Time. We discuss the efficiency of the proposed algorithm in this section. SOS and Algorithm 2 in

Table 4: Comparison of average improvement between SOS and Algorithm 2 on MeasTex-grass dataset.

	BM3D				BM3D-SAPCA			
	PSNR				PSNR			
σ	10	20	25	50	10	20	25	50
SOS	0.01	0.01	0.01	0.03	0.01	0.02	0.02	0.07
Ours	**0.01**	**0.02**	**0.05**	**0.24**	**0.01**	**0.04**	**0.06**	**0.19**
	SSIM				SSIM			
SOS	0.0005	0.0002	0.0009	0.0051	0.0001	0.0003	0.0006	0.0031
Ours	**0.0007**	**0.0003**	**0.0018**	**0.0165**	**0.0001**	**0.0008**	**0.0007**	**0.0052**
	DSI				DSI			
SOS	0.1390	0.3705	0.5502	0.3759	0.0393	0.3872	0.6542	**1.2431**
Ours	**-0.4513**	**-0.1379**	**0.3543**	**-0.3234**	**-0.1778**	**-0.4219**	**-0.1181**	4.9756

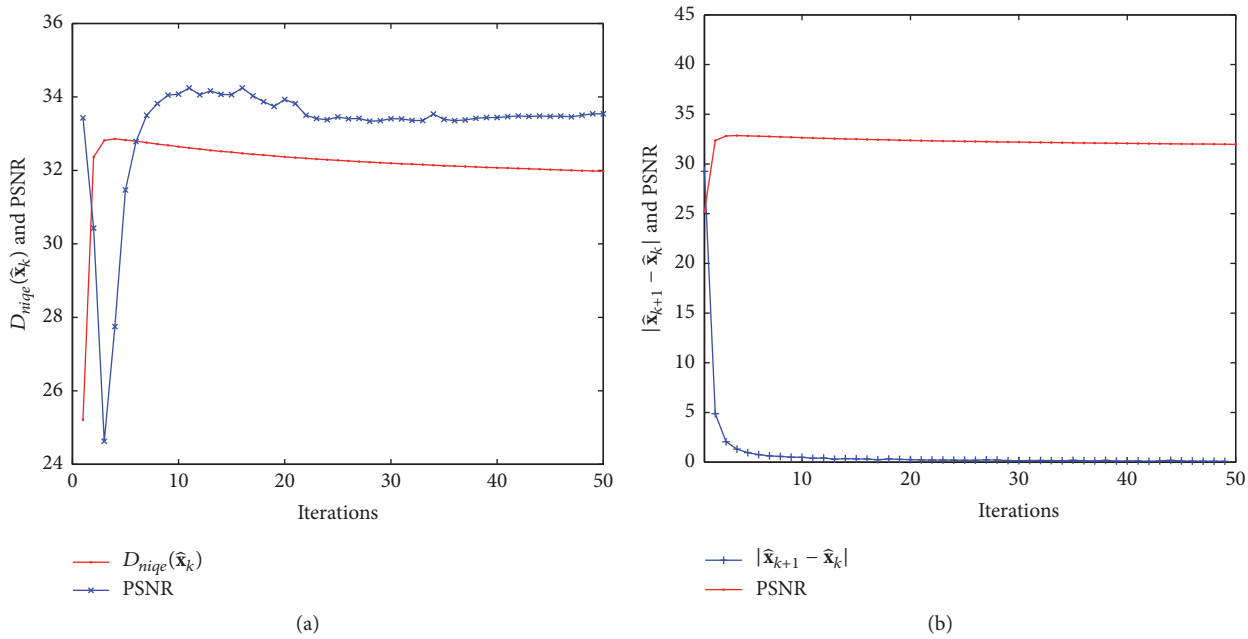
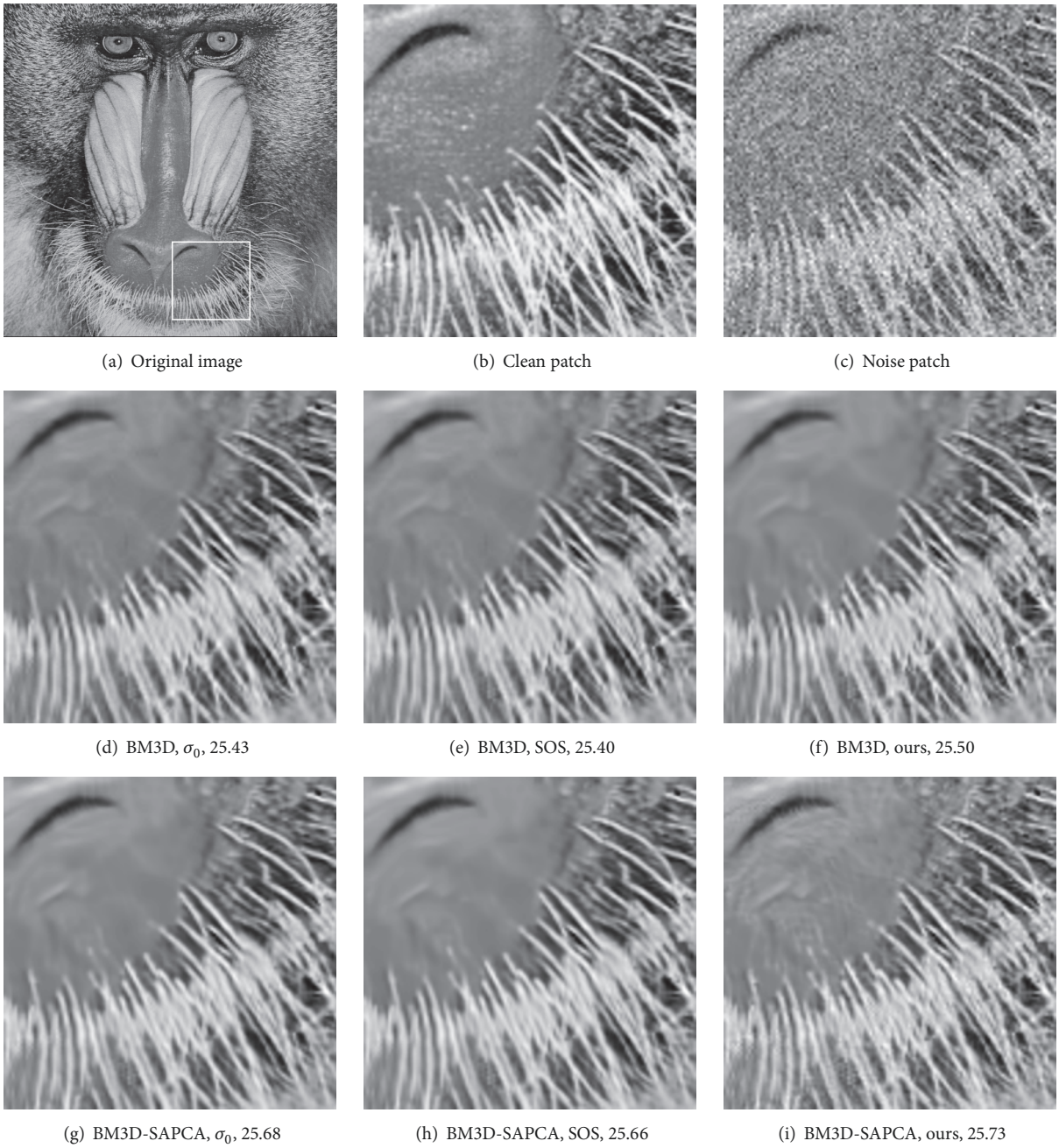


Figure 4: The comparing of stopping criterion. (a) The smallest NIQE value gives the optimal PSNR. (b) The smallest $|x_{k+1} - x_k|$ does not give the optimal PSNR.

this paper need several iterations during the denoising application because these two algorithms may cost more time than the initial denoiser. Table 5 shows the results of computational time between SOS and Algorithm 2. Compared with SOS algorithm, Algorithm 2 has a certain advantage in processing time for higher noise level while BM3D is selected as the denoiser. According to the results presented in Table 5, the processing time of Algorithm 2 in this paper is lower than SOS for each noise level when the denoiser BM3D-SAPCA is used. Furthermore, the average processing time spent per image of Algorithm 2 is less than that of SOS.

5. Conclusion and Discussion

In this paper, we proposed a new adaptive boosting denoising algorithm by plugging an accurate noise level estimation. Experience shows that the algorithm can improve performance of denoising algorithm which depended on noise level and can also keep the image's edges and detail information well. Though it is a convenient tool for improving various denoising algorithm, there are still serval directions that we are interested in future works. Firstly, in order to get optimal output image, we use NIQE algorithm to assess the quality of output image $\mathbf{x}_k$. When $D_{niqe}(\mathbf{x}_k)$ are no longer decreasing, the final result of Algorithm 2 is given by $(\mathbf{x}_{k+1})$. Figure 4(a) shows the result; the highest PSNR is obtained when $D_{niqe}(\mathbf{x}_k)$ first increase. By using Theorem 11, a straightforward stopping criterion ($|\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k|$ is sufficiently small) can be obtained. Although this straightforward stopping criterion avoid the output of Algorithm 2 back to $\mathbf{y}$, the PSNR does not always increase when $|\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k|$ decrease. Figure 4(b) shows the relationship between PSNR and $|\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k|$ with k; the smallest $|\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}_k|$ does not give the optimal PSNR. Compared with Figure 4(a) and 4(b), we can find that the stopping rule in Algorithm 2 is helpful for determining the optimal numbers of iterations. Secondly, there are two main parameters λ and



(a) Original image (b) Clean patch (c) Noise patch

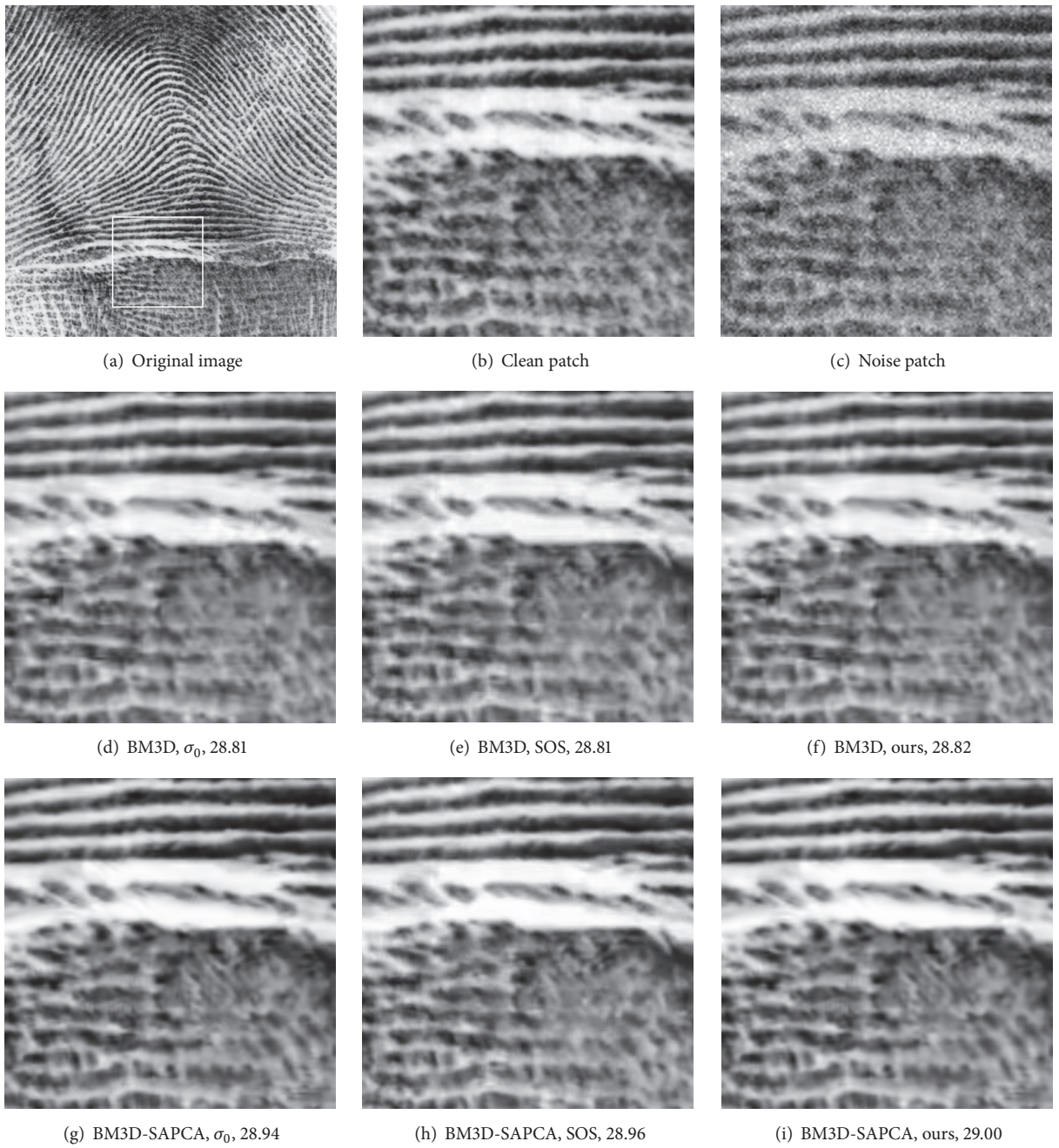
(d) BM3D, σ_0, 25.43 (e) BM3D, SOS, 25.40 (f) BM3D, ours, 25.50

(g) BM3D-SAPCA, σ_0, 25.68 (h) BM3D-SAPCA, SOS, 25.66 (i) BM3D-SAPCA, ours, 25.73

Figure 5: Visual and PSNR comparisons between standard denoising and proposed method of a 120×120 cropped region from noisy image Baboon ($\sigma = 25$). (a, b, c) are the original image and its noisy version, and (d–i) are the results when using the denoisers BM3D and BM3D-SAPCA, respectively.

Table 5: Comparison of computational time (second) between SOS and Algorithm 2.

	SIPID dataset					MeasTex-grass dataset				
	BM3D					BM3D				
σ	10	20	25	50	Ave.	10	20	25	50	Ave.
SOS	**8.47**	**8.95**	**9.05**	14.66	11.43	**15.53**	**17.22**	**17.87**	29.56	22.13
OURS	9.92	10.73	11.11	**10.89**	**10.10**	15.92	19.62	20.82	**22.68**	**19.16**
	BM3D-SAPCA					BM3D-SAPCA				
SOS	181.3	175.3	169.9	253.5	202.4	234.8	310.3	285.4	456.3	363.2
OURS	**118.0**	**112.7**	**110.8**	**133.0**	**114.3**	**268.5**	**242.9**	**228.8**	**290.6**	**248.6**



(a) Original image (b) Clean patch (c) Noise patch

(d) BM3D, σ_0, 28.81 (e) BM3D, SOS, 28.81 (f) BM3D, ours, 28.82

(g) BM3D-SAPCA, σ_0, 28.94 (h) BM3D-SAPCA, SOS, 28.96 (i) BM3D-SAPCA, ours, 29.00

Figure 6: Visual and PSNR comparisons between standard denoising and proposed method of a 120×120 cropped region from noisy image fingerprint ($\sigma = 20$). (a, b, c) are the original image and its noisy version and (d–i) are the results when using the denoisers BM3D and BM3D-SAPCA, respectively.

c in the boosting method. How to select these parameters for optimal denoising results needs further research. Finally, we hope that other image restoration problems, such as image deblurring/inpainting, can use similar embedding method.

Data Availability

It should be noted that a software release of the proposed algorithm in our manuscript is available online: https://ww2.mathworks.cn/matlabcentral/fileexchange/67924-abd.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 11671307, 61561019, 61763009, and 11761030 and by Doctoral Scientific

(a) Original image (b) Clean patch (c) Noise patch

(d) BM3D, σ_0, 30.04 (e) BM3D, SOS, 30.13 (f) BM3D, ours, 30.18

(g) BM3D-SAPCA, σ_0, 30.32 (h) BM3D-SAPCA, sos, 30.50 (i) BM3D-SAPCA, ours, 30.71

Figure 7: Visual and PSNR comparisons between standard denoising and proposed method of a 120×120 cropped region from noisy image Foreman ($\sigma = 50$). (a, b, c) are the original image and its noisy version and (d–i) are the results when using the denoisers BM3D and BM3D-SAPCA, respectively.

Fund Project of Hubei University for Nationalities under Grant No. MY2015B001.

References

[1] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *SIAM Journal on Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[2] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[3] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, vol. 6669, pp. 479–486, November 2011.

[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] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Bm3d image denoising with shape-adaptive principal component analysis," *Proc Workshop on Signal Processing with Adaptive Sparse Structured Representation Saint-malo*, 2009.

[6] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 2272–2279, October 2009.

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

[8] P. Milanfar, "A tour of modern image filtering," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 106–123, 2013.

[9] Y. Romano and M. Elad, "Improving K-SVD denoising by post-processing its method-noise," in *Proceedings of the 2013 20th IEEE International Conference on Image Processing, ICIP 2013*, pp. 435–439, 2014.

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

[11] M. R. Charest, M. Elad, and P. Milanfar, "A general iterative regularization framework for image denoising," in *Proceedings of the 2006 40th Annual Conference on Information Sciences and Systems, CISS 2006*, pp. 452–457, March 2006.

[12] N. P. Galatsanos and A. K. Katsaggelos, "Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation," *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, vol. 1, no. 3, pp. 322–336, 1992.

[13] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 2002.

[14] S. D. Chitte, S. Dasgupta, and Z. Ding, "Distance estimation from received signal strength under log-normal shadowing: Bias and variance," *IEEE Signal Processing Letters*, vol. 16, no. 3, pp. 216–218, 2009.

[15] N. Ponomarenko, V. Lukin, S. Abramov, K. Egiazarian, and J. Astola, "Blind evaluation of additive noise variance in textured images by nonlinear processing of block DCT coefficients," in *Proceedings of the SPIE - The International Society for Optical Engineering*, pp. 178–189, January 2003.

[16] V. Lukin, S. Abramov, B. Vozel, and K. Chehdi, "A method for blind automatic evaluation of noise variance in images based on bootstrap and myriad operations," *Remote Sensing. International Society for Optics and Photonics*, vol. 5982, pp. 299–310, 2005.

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

[18] M. Liu, X. 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.

[19] F. Chen, G. Zhu, and P. A. Heng, "An efficient statistical method for image noise level estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 477–485, December 2015.

[20] 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.

[21] G. Blom, "Statistical estimates and transformed beta variables," 1958.

[22] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *The Annals of Statistics*, vol. 9, no. 6, pp. 1135–1151, 1981.

[23] X. Zhu and P. Milanfar, "Automatic parameter selection for denoising algorithms using a no-reference measure of image content," *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, vol. 19, no. 12, pp. 3116–3132, 2010.

[24] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, 2012.

[25] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a completely blind image quality analyzer," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, 2013.

[26] Y. Li, S. Wang, C. Li, Z. Pan, and W. Zhang, "A fast color image segmentation approach using gdf with improved region-level ncut," *Mathematical Problems in Engineering*, vol. 2018, no. 3, Article ID 8508294, 14 pages, 2018.

[27] 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.

[28] 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.

[29] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[30] K. Egiazarian, M. Ponomarenko, V. Lukin, and O. Ieremeiem, "Statistical evaluation of visual quality metrics for image denoising," 2017.

[31] G. Smith and I. Burns, "Measuring texture classification algorithms," *Pattern Recognition Letters*, vol. 18, no. 14, pp. 1495–1501, 1997.