Information Sciences 462 (2018) 402–416

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Kernel Wiener filtering model with low-rank approximation for image denoising

Yongqin Zhang[a,b], Jinsheng Xiao[c,*], Jinye Peng[a], Yu Ding[d], Jiaying Liu[e], Zongming Guo[e], Xiaopeng Zong[b,*]

[a] School of Information Science and Technology, Northwest University, Xi'an 710127, China
[b] Department of Radiology and BRIC, University of North Carolina at Chapel Hill, NC 27599, USA
[c] School of Electronic Information, Wuhan University, Wuhan 430072, China
[d] UIH America Inc., 9230 Kirby Drive, Suite 600, Houston TX, 77054, USA
[e] Institute of Computer Science and Technology, Peking University, Beijing 100871, China

ARTICLE INFO

Article history:
Received 17 July 2017
Revised 7 June 2018
Accepted 9 June 2018
Available online 15 June 2018

Keywords:
Image denoising
Sparse representation
Low-rank approximation
Wiener filtering
Color space

ABSTRACT

Sparse representation and low-rank approximation have recently attracted great interest in the field of image denoising. However, they have limited ability for recovering complex image structures due to the lack of satisfactory local image descriptors and shrinkage rules of transformed coefficients, especially for degraded images with heavy noise. In this paper, we propose a novel kernel Wiener filtering model with low-rank approximation for image denoising. In the model, a shape-aware kernel function is introduced to describe local complex image structures. The reference image of kernel Wiener filtering is estimated by an optimized low-rank approximation approach, where eigenvalue thresholding is deduced for the shrinkage of transformed coefficients using *a prior* nonlocal self-similarity. Finally the optimal kernel Wiener filter is derived for image noise reduction. Our experimental results show that the proposed model can faithfully restore detailed image structures while removing noise effectively, and often outperforms the state-of-the-art methods both subjectively and objectively.

1. Introduction

Due to imperfect devices, intrinsic thermal fluctuations, and external interference, data acquisition and transmission inevitably introduce noises into captured images. Image denoising is an essential tool for recovering the underlying clean images from the noisy observations. Despite continuous research efforts over the past four decades, it is still a challenging task to develop effective and robust image denoising techniques that can be applied to a wide range of real-world scenarios.

The existing image denoising methods can be roughly classified into two main categories: spatial-domain methods and transformed-domain methods. The first category includes partial differential equation (PDE)-based methods [28,31,39,40], spatially varying convolution methods [26,29], and nonlocal means-based methods [3,7,12,14,15,19,37]. In PDE-based methods, a total variation (TV) filter [28,31,39,40] is often introduced to remove noise while preserving edges. However, TV

* Corresponding authors.
E-mail addresses: zhangyongqin@pku.edu.cn (Y. Zhang), xiaojs@whu.edu.cn (J. Xiao), pjy@nwu.edu.cn (J. Peng), yuding99@yahoo.com (Y. Ding), liujiaying@pku.edu.cn (J. Liu), guozongming@pku.edu.cn (Z. Guo), zongxp@gmail.com (X. Zong).

https://doi.org/10.1016/j.ins.2018.06.028
0020-0255/

assumes piecewise smoothness of underlying images, which often cannot be satisfied for natural images and results in undesired staircase effect and loss of structural details. The spatially varying convolution methods convolve a noisy image with a pointwise local geometry-driven kernel function to remove noise while preserving local structures. However, the selection of optimal kernel functions is often challenging for such methods. Different from the above-mentioned local smoothing filters, the nonlocal means-based methods first introduced by Buades et al. [3] exploit the nonlocal self-similarity of spatial patterns within natural images for noise removal. However, such method ignores *a prior* knowledge on local image structures.

Compared to the above spatial-domain methods, noise and signal can often be more easily separated in a transformed domain. Thus, transformed-domain denoising methods have received great attention in recent years, which include wavelet transform-based [7,13,18,20,27], sparse representation-based [5,9,11,15,16,33,34,37,38], and deep learning-based methods [6,8,36]. The wavelet transform-based methods [27] decompose an input noisy image into multiple scales and then wavelet coefficients at each scale are shrunk towards zero to suppress noise in the transform domain. However, as the wavelet basis is fixed, such methods cannot adaptively represent various structural patterns of natural images, which inevitably lead to noticeable visual artifacts. Sparse representation-based methods overcome this issue by learning data-adaptive bases for image denoising [9,10,15,21,37]. However, these methods generally involve solving a complex optimization problem and manually tuning parameters to approach the optimal performance. To overcome the limitations of model-based methods mentioned above, deep learning-based methods learn image models from big training data and then use the trained models for image denoising. Such methods can achieve highly accurate results with high computational efficiency [23,24]. However, these methods need large training data and often produce a significant drop in denoising performance when there are discrepancies between test and training images.

In this paper, we propose a novel kernel Wiener filtering model (KWFM) to faithfully restore detailed image structures while further improving the image denoising performance. The reference images are estimated by an optimized low-rank approximation approach. The main contributions of this work are threefold: 1) The shape-adaptive kernel function is introduced as a local image descriptor to construct the optimal kernel Wiener filter with minimum mean square error; 2) The constrained covariance matrix minimization is modeled to estimate the reference image by optimized low-rank approximation (OLRA); 3) Eigenvalue thresholding is derived for the shrinkage of transformed coefficients to reduce the estimation bias of the low-rank approximation. The proposed algorithm was evaluated in comparison with the current popular methods both qualitatively and quantitatively on numerous test images.

The remainder of this paper is organized as follows. Section 2 presents the overview of our KWFM and elaborates the model solutions with an emphasis on the closed-form or convergent solutions to its subproblems. Section 3 shows the experimental results. The discussions are given in Section 4 and conclusions are drawn in Section 5.

2. Method

Throughout this paper, we denote scalars, vectors and matrices by non-boldfaced, boldfaced lower-case, and boldfaced upper-case letters, respectively.

2.1. Image degradation model

In the presence of noise, the observed images in denoising studies are often represented by the following simplified model [3]:

$$\mathbf{Y} = \mathbf{X} + \upsilon, \tag{1}$$

where $\mathbf{Y}$ is the observed image, $\mathbf{X}$ is the clean image, and υ represents Gaussian noise. The goal of image denoising is to obtain the best estimate of the clean image $\mathbf{X}$ from the observed noisy image $\mathbf{Y}$.

2.2. Kernel Wiener filtering model

The images generally contain complex structures, such as edges, textures and smooth regions. These structures are corrupted by noise in the noisy images. However, the existing denoising methods often fail to recover detailed image structures while removing noise. The conventional Wiener filtering (CWF) as the second stage of BM3D [7] is a commonly used method to further improve the denoising performance, but it may often fail when integrated with the best state-of-the-art methods, e.g. WNNM [15]. To faithfully restore detailed image structures while further improving the denoising performance, we propose a novel kernel Wiener filtering (KWF) model by incorporating the shape-aware kernel function to represent local complex image structures. Let $\mathbf{y}_i = \mathbf{C}_i\mathbf{Y}$ and $\mathbf{r}_i = \mathbf{C}_i\mathbf{R}$ denote an image patch extracted from the noisy image $\mathbf{Y}$ and the reference image $\mathbf{R}$ at pixel i, respectively, where $\mathbf{C}_i$ is a matrix extracting a patch from the image at pixel i. Then the proposed KWFM method can be formulated as follows:

$$\hat{\mathbf{X}} = \left(\sum\nolimits_i \mathbf{C}'_i\mathbf{C}_i\right)^{-1} \sum\nolimits_i \mathbf{C}'_i\left(\hat{\mathbf{h}}_i * (\mathbf{k}_i \circ \mathbf{y}_i)/\mathbf{k}_i\right), \tag{2}$$

where $\hat{\mathbf{X}}$ is the denoised image regarded as the estimate of the clean image $\mathbf{X}$, $\mathbf{C}'_i$ is the transpose matrix of $\mathbf{C}_i$, $\hat{\mathbf{h}}_i$ of size $L \times L$ is the optimal kernel Wiener filter $\mathbf{H}$ at pixel i, $*$ denotes the convolution operation, $\circ$ denotes the Hadamard product,

/ represents the Hadamard division, and $\mathbf{k}_i$ is the shape-aware kernel function $\mathbf{K}$ at pixel i that implements data mapping from the signal space to its feature space. The optimal kernel Wiener filter $\hat{\mathbf{h}}_i$ in Eq. (2) is obtained as:

$$\begin{aligned} &\hat{\mathbf{h}}_i = \min E\left[e_i^2(j)\right], \\ &\text{with} \quad \mathbf{e}_i = \mathbf{k}_i \circ \mathbf{r}_i - \mathbf{h}_i * (\mathbf{k}_i \circ \mathbf{y}_i), \end{aligned} \tag{3}$$

where $E[\cdot]$ denotes the expected value, and $e_i(j)$ is the j-th element of the patch error $\mathbf{e}_i$. Considering the estimation bias of the conventional low-rank estimators in obtaining the reference images [1], e.g. singular value thresholding (SVT) [4,9], we propose an OLRA approach based on the minimization of constrained covariance matrix nuclear norm to increase the estimation accuracy of the reference images. Specifically, for each target patch of an input noisy image, the structural similarity is exploited to construct a group of similar patches. Then noise in each group of similar patches is reduced by eigenvalue thresholding. The resulting patches are collected to estimate the reference image of KWF by the weighted averaging method.

For grayscale images, the proposed method can be applied directly to the noisy input images. For color images with red-green-blue (RGB) color components, the three components are typically highly correlated. Therefore, an RGB image is first transformed into the decorrelated YUV image in terms of one luminance (Y) and two chrominance components (U and V) by forward color space conversion. Then the proposed method are separately applied to each of the decorrelated YUV images. Finally, the denoised RGB images are reconstructed from the denoised YUV images by inverse color space conversion.

2.3. KWFM algorithm implementation

The KWFM algorithm consists of first estimating the reference image via optimized low-rank approximation, then design the kernel Wiener filter via least square minimization, and finally convoluting the kernel Wiener filter with the kernelized noisy images to obtain the denoised image.

2.3.1. Optimized low-rank approximation

In the first stage, we need to get an accurate estimate of the reference images for the design of the optimal kernel Wiener filter. For each target patch $\mathbf{y}_i$ of size $p \times p$ centered at pixel i in the noisy image $\mathbf{Y} \in \mathbb{R}^{M \times N}$, the similar patches of $\mathbf{y}_i$ are found across a sliding search window of size $W \times W$ in the noisy image $\mathbf{Y}$ by the block-matching method [7,15,37]. These similar patches are collected to form a reshaped matrix $\mathbf{G}_i = \left[\mathbf{y}_{i,1}, \mathbf{y}_{i,2}, \ldots, \mathbf{y}_{i,Q}\right] \in \mathbb{R}^{p^2 \times Q}$, where Q is the number of similar patches. The noise model in Eq. (1) can be rewritten in the patch-based representation as follows:

$$\mathbf{G}_i = \mathbf{O}_i + \varsigma_i, \tag{4}$$

where $\mathbf{G}_i$, $\mathbf{O}_i$ and ς_i denote the patch matrices from the noisy image $\mathbf{Y}$, the clean image $\mathbf{X}$ and the noise υ, respectively.

In the past decade, the singular value soft thresholding method [4,9,15] has been widely studied for the low-rank approximation problem. However, the estimation bias of the soft thresholding affects the accuracy of the reconstructed signals [1]. On the other hand, the discontinuity of the hard thresholding may cause oscillation of the reconstructed signals. It was observed that the region covariance descriptor achieves superior performance in object detection and classification [30]. To estimate the singular values accurately in the low-rank approximation, we propose an OLRA method based on the constrained nuclear norm minimization by introducing the structured sparsity [38] and covariance matrices [2]. OLRA estimates each patch group $\mathbf{O}_i$ in the reference image $\mathbf{R}$ as follows:

$$\begin{aligned} &\hat{\mathbf{O}}_i = \min_{\mathbf{O}_i} \left\| \mathbf{O}_i \mathbf{O}_i' \right\|_*, \\ &\text{s.t.} \quad \|\mathbf{G}_i - \mathbf{O}_i\|_F^2 < \xi, \end{aligned} \tag{5}$$

where the nuclear norm $\|\cdot\|_*$ is equivalent to the group sparsity constraint [4,9], $\|\cdot\|_F$ is the Frobenius norm, and ξ is an error constraint. In implementation, ξ is not set explicitly. Instead, Eq. (5) is solved iteratively to estimate the reference image $\mathbf{R}$. The iteration is terminated when the iteration number τ reaches a maximum value T. The value of ξ in Eq. (5) is related to the total number of iterations T. At each iteration τ, the eigenvalue coefficients of each similar patch group $\mathbf{G}_i^{(\tau)}$ are sequentially shrunk to minimize the representation error by SVD. Specifically, at the beginning of each iteration τ, the noisy image $\mathbf{Y}^{(\tau)}$ and its noise deviation σ_τ are separately updated as follows:

$$\mathbf{Y}^{(\tau)} = \hat{\mathbf{R}}^{(\tau)} + \varepsilon\left(\mathbf{Y} - \hat{\mathbf{R}}^{(\tau)}\right), \tag{6}$$

$$\sigma_\tau = \sqrt{\max\left(\sigma_\upsilon^2 - \sigma_{\text{RMSE}}^2, 0\right)}, \tag{7}$$

where ε is a control parameter, σ_υ^2 is the noise variance of the input noisy image $\mathbf{Y}$, and σ_{RMSE}^2 is the mean squared error between $\mathbf{Y}$ and $\mathbf{Y}^{(\tau)}$. Note that $\mathbf{Y}$ is used as the initial estimate $\hat{\mathbf{R}}^{(\tau)}$ of the reference image at $\tau = 0$. By using the iterative projected gradient-descent algorithm [4,17], Eq. (5) is solved with the eigenvalue thresholding (EVT) for shrinking the eigenvalue coefficients $\mathbf{\Sigma}_i^{(\tau)}$ of $\mathbf{G}_i^{(\tau)}$ as follows:

$$\hat{\mathbf{\Sigma}}_i^{(\tau)} = \sqrt{\max\left(\left(\mathbf{\Sigma}_i^{(\tau)}\right)^2 - Q\sigma_\tau^2, 0\right)}, \tag{8}$$

where the shrinkage amount $Q\sigma_\tau^2$ is adopted based on the relationship between the eigenvalues for the noisy and clean images as provided in the Appendix A. Our EVT has higher estimation accuracy with less bias than the soft thresholding used in SAIST [9] for signal recovery, as shown in the *Experimental Results* Section.

Let $\mathbf{U}_i^{(\tau)}$ and $\mathbf{V}_i^{(\tau)}$ denote a $p^2 \times p^2$ unitary matrix and a $Q \times Q$ unitary matrix in the SVD transform of $\mathbf{G}_i^{(\tau)}$, respectively. The shrunk singular values $\hat{\mathbf{\Sigma}}_\mathbf{i}^{(\tau)}$ are calculated from Eq. (8), and then used to construct the denoised similar patch group by the inverse SVD transform as follows:

$$\hat{\mathbf{O}}_i^{(\tau+1)} = \mathbf{U}_i{}^{(\tau)} \hat{\mathbf{\Sigma}}_i^{(\tau)} \mathbf{V}_i{}'^{(\tau)}. \tag{9}$$

At the end of each iteration, as the different groups of similar patches containing an identical pixel may have different ranks, an estimation bias of the pixel will be produced in the reconstructed image. To improve the estimation accuracy of the ideal reference image $\mathbf{R}$, we employ a weighted averaging method to reconstruct the whole estimated image $\hat{\mathbf{R}}^{(\tau+1)}$ by aggregating all denoised patch groups $\hat{\mathbf{O}}_i^{(\tau+1)}$. Like BM3D [7] and APCAS [37], we assign an empirical weight to each denoised patch group $\hat{\mathbf{O}}_i^{(\tau+1)}$. The empirical weight w_i is given as

$$w_i = \begin{cases} 1 - Q_\tau/Q, & Q_\tau < Q \\ 1/Q, & Q_\tau = Q \end{cases}, \tag{10}$$

where Q_τ is the estimated rank of each patch group matrix $\mathbf{G}^{(\tau)}$. Finally, the whole estimated reference image is built by the weighted averaged method as follows:

$$\hat{\mathbf{R}}_k^{(\tau+1)} = \left(\sum_{i,j} w_i \hat{\mathbf{O}}_{i,j,k}^{(\tau+1)} \right) \Big/ \left(\sum_{i,j} w_i \right), \tag{11}$$

where R_k refers to pixel k in the reference image, and $\hat{\mathbf{O}}_{i,j,k}$ refers to the denoised image intensity of pixel k in the jth patch of patch group i, and the summation is carried out over all patches that overlap with pixel k.

The above procedure is repeated until the convergence condition (e.g., the maximum number of iterations) is achieved. The final estimated reference image $\hat{\mathbf{R}}$ is used to design the optimal kernel Wiener filter at the second stage of our KWFM algorithm.

2.3.2. Kernel Wiener filtering

As mentioned above, CWF performance often deteriorates when combined with state-of-the-art denoising methods (e.g., WNNM [15]). To restore complex image structures, such as edges and textures, Yamashita et al. [35] proposed a KWF method by using the first-order approximation of the Gaussian kernel function. But they [35] did not further investigate the optimization of KWF, such as the kernel shape and the reference image estimation. To further improve the KWF, we introduce the shape-aware kernel function to represent complex image structures, and estimate the reference image $\mathbf{R}$ by the OLRA described above. Incidentally, our KWF amounts to the dual-domain filtering [21,22], except that a different shrinkage function is applied to the Fourier coefficients.

Since the bilateral function [29] can adaptively represent the geometrical shape of local image regions effectively, we adopt the bilateral function as the shape-aware kernel function $\mathbf{K}$, which is based on the similarity measure between the target pixel i and its neighboring pixel q in the reference image $\mathbf{R}$. Specifically, the shape-aware bilateral kernel centered at pixel i is constructed from the reference image as follows:

$$k_{i,q} = \exp\left(-\frac{|i-q|^2}{2\sigma_s^2} \right) \exp\left(-\frac{(r_i - r_q)^2}{\beta_r \sigma_v^2} \right), \tag{12}$$

where r_i and r_q are the intensity of pixels i and q, respectively. σ_v^2 is the noise variance of the input noisy image $\mathbf{Y}$. σ_s and β_r are smoothing parameters controlling the Gaussian functions in spatial and intensity domains, respectively. For visual illustration, Fig. 1 shows the three-dimensional shaded surface of the shape-aware bilateral kernel function on a test image.

With the shape-aware bilateral kernel defined in Eq. (12) and estimated reference images, the kernel Wiener filter can be obtained from Eq. (3). To reduce the high computational cost, we can use the Fourier transform to calculate the kernel Wiener filter. Specifically, for each pixel i and its noisy patch $\mathbf{y}_i$ of size $L \times L$ centered at pixel i in the noisy image $\mathbf{Y} \in \mathbb{R}^{M \times N}$, the element-wise product of the corresponding estimated reference patch $\hat{\mathbf{r}}_i$ and the related kernel patch $\mathbf{k}_i$ is first converted into Fourier coefficients by the forward Fourier transform. According to the Wiener-Hopf equation, the kernel Wiener filter in Fourier domain for each patch $\mathbf{y}_i$ of the noisy image $\mathbf{Y}$ can be expressed as

$$\mathcal{F}\left(\hat{\mathbf{h}}_i\right) = \frac{\left|\mathcal{F}\left(\mathbf{k}_i \circ \hat{\mathbf{r}}_i\right)\right|^2}{\left|\mathcal{F}\left(\mathbf{k}_i \circ \hat{\mathbf{r}}_i\right)\right|^2 + \beta_f \sigma_i^2}, \tag{13}$$

where $\mathcal{F}(\cdot)$ denotes the discrete Fourier transform (DFT), β_f is the shrinkage parameter, and σ_i^2 is the variance of Fourier coefficients of the kernel patch $\mathbf{k}_i$:

$$\sigma_i^2 = \sigma_v^2 \sum_{q \in \Omega_i} k_{i,q}^2, \tag{14}$$

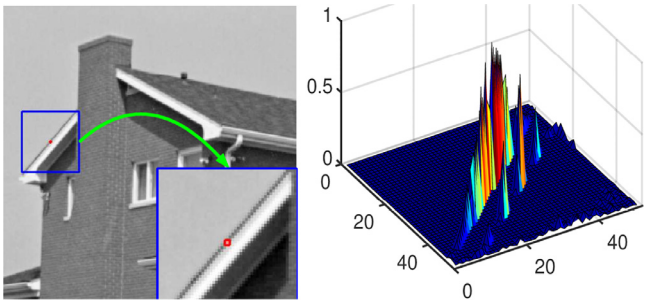


Fig. 1. Three-dimensional shaded surface of the shape-aware bilateral kernel function centered at a target pixel from the test image *'house'*. The target pixel is denoted in red, while its local region is labeled in the blue box.

Algorithm 1 Pseudocodes of the KWFM Algorithm

Input: a noisy image $\mathbf{Y}$.
Output: a denoised image $\hat{\mathbf{X}}$.

I. Initialize the parameters T, Q, ε, L, σ_s, β_r and β_f;
II. Perform the forward color space transformation;
III. OLRA: Initialize $\hat{\mathbf{R}}^{(0)} = \mathbf{Y}$, and then for each iteration $\tau = 0$ to T do

- Update $\mathbf{Y}^{(\tau)}$ and σ_τ using Eqs. (6) and (7);
- For each patch $\mathbf{y}_i^{(\tau)}$ in $\mathbf{Y}^{(\tau)}$ do

  1) Find $\mathbf{G}^{(\tau)}$ by the block-matching search;
  2) Compute the SVD transform of $\mathbf{G}^{(\tau)}$;
  3) Shrink the eigenvalue coefficients using Eq. (8);
  4) Compute $\hat{\mathbf{O}}_i^{(\tau+1)}$ by the inverse SVD;

- Update $\hat{\mathbf{R}}^{(\tau+1)}$ by Eqs. (10) and (11);

IV. KWF: for each target patch $\mathbf{y}_i$ centered at pixel i in $\mathbf{Y}$ do

- Compute the kernel $\mathbf{k}_i$ from $\hat{\mathbf{R}}$ using Eq. (12);
- Compute σ_i^2, $\bar{y}_i$, $\bar{r}_i$, $\tilde{\mathcal{H}}_{i,f}$, and $\tilde{\mathcal{Y}}_{i,f}$;
- Estimate each pixel value $\hat{x}_i$ of $\hat{\mathbf{X}}$ using Eq. (15);

V. Construct the denoised image $\hat{\mathbf{X}}$.

where Ω_i denotes a coordinate set of neighboring pixels around the pixel i.

Then, for each pixel i and its noisy patch $\mathbf{y}_i$ of size $L \times L$ in the noisy image $\mathbf{Y}$, the corresponding denoised pixel value $\hat{x}_i$ is obtained from the Fourier coefficients of $\mathbf{y}_i$ over the frequency domain as follows:

$$\hat{x}_i = \frac{1}{L^2} \sum_{f \in \Gamma_i} \hat{\mathcal{H}}_{i,f} \mathcal{Y}_{i,f}, \tag{15}$$

where Γ_i denotes the local frequency domain, f is a coordinate in the frequency domain, $\hat{\mathcal{H}}_{i,f}$ denotes the Fourier transform of $\hat{\mathbf{h}}_i$, and $\mathcal{Y}_{i,f}$ is the Fourier transform of $\mathbf{k}_i \circ \mathbf{y}_i$.

Finally, the whole denoised grayscale image is constructed by aggregating all the denoised pixels. For color image denoising, after removing noise of each channel of the decorrelated color image, the denoised color image is obtained by inverse color space conversion. The proposed KWFM algorithm is summarized in Algorithm 1.

3. Experimental results

In this section, we first present the test data and parameter settings of our KWFM algorithm. Then we assess the impact of each KWFM stage on the denoising performance and compare KWFM with competing methods.

Table 1
Chosen parameter values of the KWFM algorithm. Here the symbol 'G/C' is the abbreviation of grayscale/color images.

Noise level	p	Q	T	β_r (G/C)	β_f (G/C)
$\sigma_\upsilon \leqslant 20$	6	70	8	0.81/2.2	1.6/0.8
$20 < \sigma_\upsilon \leqslant 40$	8	90	10	0.32/2.0	1.4/1.0
$40 < \sigma_\upsilon \leqslant 60$	8	105	14	0.28/1.1	1.2/1.2
$\sigma_\upsilon > 60$	9	130	14	0.24/0.6	1.0/1.2

Table 2
Averaged PSNR (dB) and SSIM results of these different methods for the grayscale image dataset 'BSD68' with various noise levels. The best results are highlighted in boldface.

Noise level	BM3D [7]	SAIST [9]	DDID2 [22]	WNNM [15]	DnCNN-S [36]	WCWF	OLRA	KWFM
10	33.32/.9164	33.44/.9176	33.37/.9196	33.59/.9205	**33.88/.9270**	33.50/.9199	33.60/.9212	33.62/.9233
30	27.76/.7736	27.82/.7731	27.90/.7740	27.99/.7817	**28.36/.7999**	27.90/.7801	27.99/.7829	28.09/.7913
50	25.62/.6869	25.65/.6868	25.75/.6827	25.87/.6989	**26.23/.7189**	25.76/.6973	25.85/.6946	26.02/.7080
70	24.44/.6331	24.46/.6396	24.46/.6203	24.64/.6466	**24.90/.6567**	24.52/.6452	24.56/.6384	24.77/.6489

3.1. Dataset

To verify the performance of KWFM both subjectively and objectively, we implemented numerous experiments on some widely used test images chosen from publicly available datasets [25][1,2]. The test images consisted of two grayscale image datasets (i.e., Set11 and BSD68 [6,36]) and two color image datasets (i.e., Set12 and CBSD68 [6,36]). According to the image degradation model defined in Eq. (1), these datasets were synthetically corrupted by the additive white Gaussian noise (AWGN) with different noise levels to simulate the noisy observations.

3.2. Experimental settings

The convergence condition is defined as the iteration number equals to a predetermined maximum number. The parameters for OLRA (ε, W, p, Q and T) were manually tuned to get an accurate estimate of the reference image. Then, the other four parameters L, σ_s, β_r and β_f were empirically estimated to ensure that the proposed method achieves approximately the best performance. In our experiments, the basic parameters of KWFM were as follows: $\varepsilon = 0.1$, $W = 61$, $L = 83$, $\sigma_s = 20$ and the other parameters are shown in Table 1. To verify the robustness of OLRA and KWF against the changes of these parameters, we selected one test image *'Montage'* and fixed the noise deviation to $\sigma_\upsilon = 50$. By changing a parameter while keeping all other parameters at their current values, we calculated the PSNR values of the reference images and denoised images as a function of the parameter value. The results are given in Fig. 2. Fig. 2 shows that the parameters p, Q, ε and β_r have a strong influence on the PSNR values, while the other parameters only change the PSNR value by < 0.2 dB. As T increases, PSNR of OLRA initially increases but then converges to a constant value c. The impact of these parameters on other images and noise levels are similar to the PSNR plots in Fig. 2.

3.3. Results for grayscale images

We first carried out an experiment to separately assess the impacts of the two stages of the proposed method. For quantitative evaluation, two image quality metrics: the Structural SIMilarity (SSIM) [32] and PSNR, were used to measure the similarity between the denoised image and the original clean image. SSIM [32] is more reliable than PSNR for visual quality assessment. In this experiment, we calculated the PSNR and SSIM of the images output from the two stages of our KWFM methods (labeled as 'OLRA' and 'KWFM', respectively), and compared them with with WNNM [15] and a united framework (denoted as WCWF) consisting of two successive stages: WNNM [15] and CWF, where the output of WNNM [15] is used as the reference image of CWF. Table 2 and Table 3 give the resulting image quality metrics for two sets of grayscale images (i.e., BSD68 and Set11), respectively. Fig. 3 gives a visual comparison of the denoised images for two test images *'House'* and *'Montage'* corrupted by the Gaussian noise with standard deviations of 50 and 100, respectively. Table 2 to 3 and Fig. 3 show that while WCWF often has better SSIM and visual results than WNNM [15], WCWF is mostly worse than WNNM [15] in PSNR, indicating that CWF as the second stage of WCWF, BM3D [7] and APCAS [37] is not suitable for improving the performance of image denoising. Furthermore, although our OLRA sometimes achieves higher PSNR values than WCWF and WNNM [15], it usually has worse SSIM values and visual results than WCWF and WNNM [15]. However, our KWFM generally achieves better PSNR/SSIM results and less visual artifacts than OLRA, WCWF and WNNM

[1] Allan Weber, The USC-SIPI Image Database, March 31, 2015, http://sipi.usc.edu/database/.
[2] Rich Franzen, Kodak Lossless True Color Image Suite, March 31, 2013, http://r0k.us/graphics/kodak/.

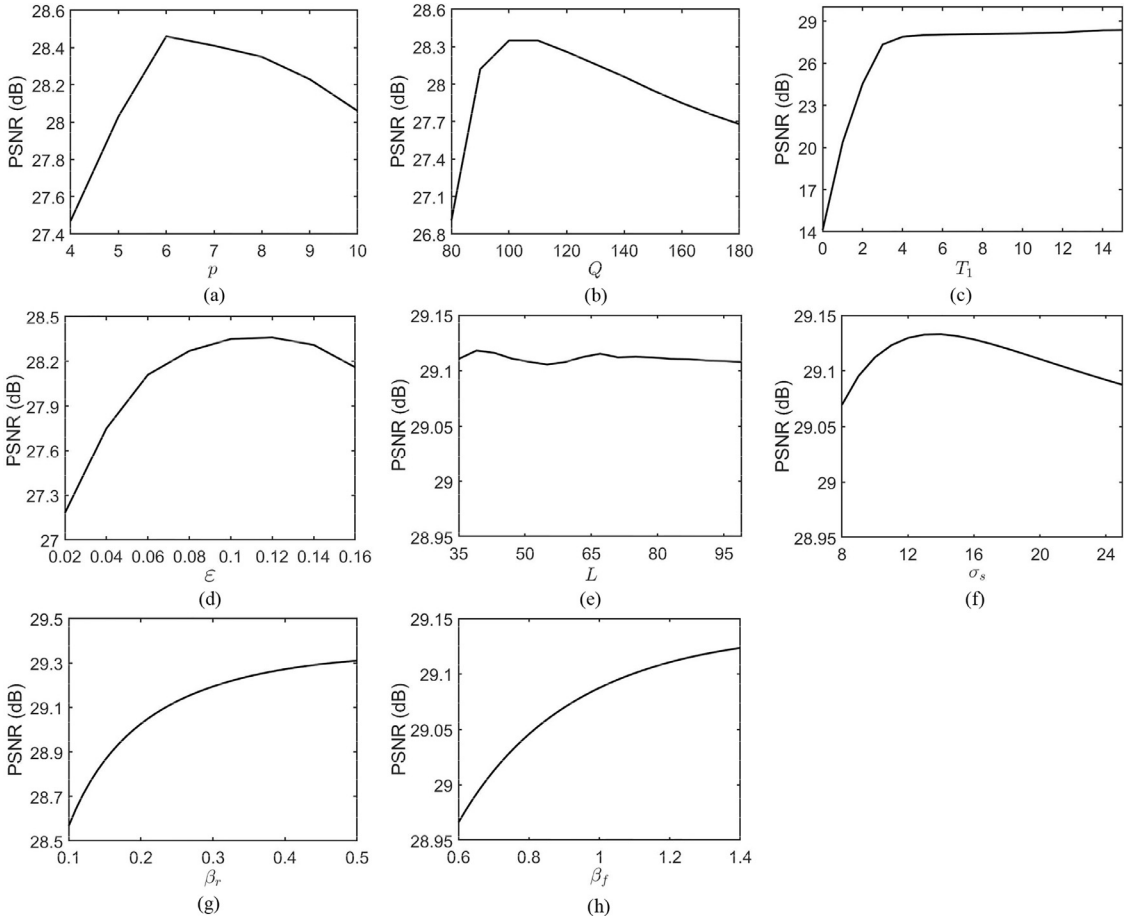
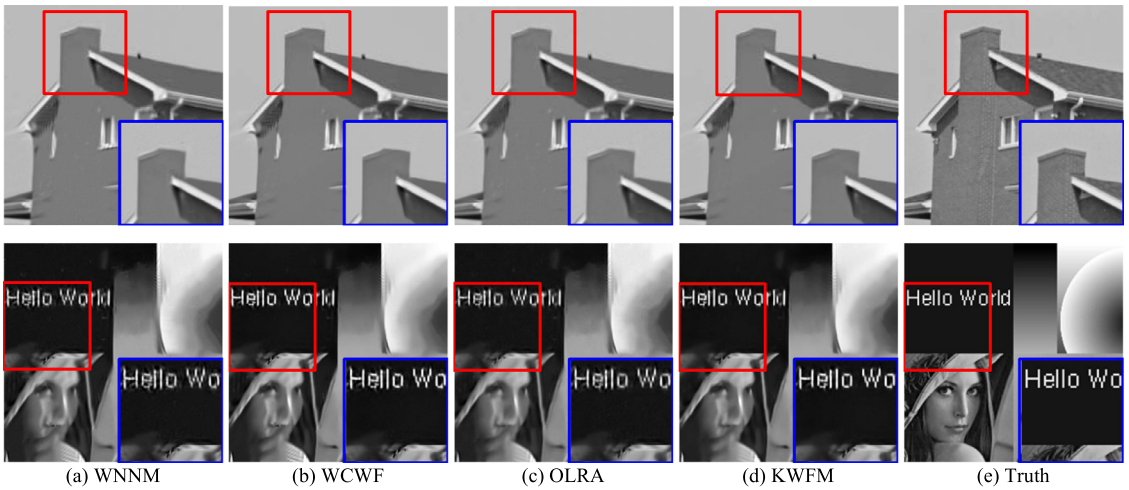


Fig. 2. PSNR values of the denoised images separately obtained by our OLRA and KWF against the changes of key parameters for the image *'Montage'* corrupted by the Gaussian noise with standard deviation 50. (a)∼(d) for OLRA, and (e)∼(h) for KWF.



(a) WNNM (b) WCWF (c) OLRA (d) KWFM (e) Truth

Fig. 3. Visual comparison of the denoised results obtained by WNNM [15] and our proposed methods. The grayscale image *'House'* is corrupted by the Gaussian noise with standard deviation 50 in the top row, whereas the grayscale image *'Montage'* is corrupted by the Gaussian noise with standard deviation 100 in the bottom row. From left to right: (a) Truth, (b) WNNM [15] (Top: PSNR (dB)/SSIM = 30.32/.8222, Bottom: PSNR (dB)/SSIM = 24.16/.7766), (c) WCWF (Top: PSNR (dB)/SSIM = 30.13/.8241, Bottom: PSNR (dB)/SSIM = 24.06/.7739), (d) OLRA (Top: PSNR (dB)/SSIM = 30.32/.8163, Bottom: PSNR (dB)/SSIM = 24.21/.7830), and KWFM (Top: PSNR (dB)/SSIM = **30.56/.8305**, Bottom: PSNR (dB)/SSIM = **24.93/.8136**). CWF often deteriorates the performance when combined with state-of-the-art denoising methods including WNNM [15], while our KWF still exhibits superior robustness and denoising performance.

Table 3
PSNR (dB) and SSIM results of these different methods for the grayscale image dataset 'Set11' with various noise levels. The best results are highlighted in boldface.

Noise level		BM3D [7]	SAIST [9]	DDID2 [22]	WNNM [15]	DnCNN-S [36]	WCWF	OLRA	KWFM
Barbara	10	34.98/.9420	35.24/.9425	34.70/.9417	**35.51/.9448**	34.60/.9399	35.32/.9438	35.46/.9447	35.31/.9436
	30	29.81/.8687	30.14/.8781	29.99/.8791	30.31/.8810	28.92/.8524	30.27/.8816	30.34/.8824	**30.37/.8864**
	50	27.23/.7946	27.51/.8040	27.47/.8062	27.79/.8199	26.22/.7693	**27.82**/.8231	27.58/.8080	27.77/**.8237**
	70	25.47/.7261	25.87/.7487	25.78/.7400	26.17/.7631	23.97/.6680	26.16/**.7656**	26.00/.7534	**26.19**/.7655
Boats	10	33.92/.8878	33.91/.8860	33.84/.8880	34.09/.8900	34.09/.8897	34.00/.8877	**34.12/.8909**	34.05/.8890
	30	29.12/.7795	28.98/.7696	29.07/.7731	29.24/.7803	**29.38/.7853**	29.21/.7825	29.26/.7812	29.32/.7841
	50	26.78/.7053	26.63/.6922	26.73/.6923	26.97/.7083	**27.20/.7188**	26.93/.7130	26.95/.7034	27.12/.7144
	70	25.40/.6526	25.28/.6469	25.27/.6342	25.57/.6573	**25.77/.6647**	25.52/.6622	25.46/.6514	25.64/.6588
C.man	10	34.18/.9319	34.30/.9345	34.22/.9340	34.44/.9332	**34.68/.9372**	34.32/.9334	34.49/.9340	34.53/.9368
	30	28.64/.8375	28.36/.8253	28.86/.8352	28.80/.8401	**29.27/.8573**	28.68/.8417	28.76/.8409	28.99/.8511
	50	26.12/.7824	26.14/.7765	26.50/.7779	26.42/.7842	**27.02/.8058**	26.27/.7856	26.46/.7784	26.78/.7984
	70	24.61/.7424	24.58/.7371	24.84/.7287	24.85/.7439	**25.40**/.7641	24.75/.7471	24.82/.7453	25.24/**.7653**
Couple	10	34.04/.9094	33.96/.9047	33.99/.9075	34.14/.9101	**34.32/.9138**	34.08/.9098	34.19/.9109	34.21/.9112
	30	28.87/.7947	28.72/.7820	28.80/.7849	28.98/.7953	**29.21/.8034**	28.97/.7986	28.99/.7965	29.11/.8014
	50	26.46/.7068	26.30/.6943	26.32/.6894	26.65/.7136	**26.90/.7249**	26.66/.7199	26.61/.7084	26.81/.7213
	70	25.00/.6406	24.87/.6352	24.75/.6153	25.17/.6520	**25.30/.6546**	25.18/.6598	25.03/.6428	25.25/.6538
F.print	10	32.46/.9688	32.69/.9699	31.88/.9661	32.81/.9708	32.62/.9700	32.74/.9706	**32.82/.9709**	32.69/.9706
	30	26.83/.8936	26.95/.8935	26.47/.8824	26.99/.8961	26.59/.8915	26.99/.8970	**27.02**/.8968	26.99/**.8992**
	50	24.53/.8308	24.52/.8254	24.17/.8120	**24.67**/.8353	24.10/.8217	24.65/**.8375**	24.51/.8281	24.54/.8339
	70	23.12/.7802	23.17/.7787	22.79/.7557	**23.31**/.7890	22.39/.7497	23.26/**.7928**	23.08/.7732	23.11/.7769
Hill	10	33.62/.8834	33.65/.8831	33.66/.8842	33.79/.8869	**33.86/.8903**	33.71/.8856	33.83/.8879	33.85/.8888
	30	29.16/.7504	29.06/.7409	29.07/.7395	29.25/.7514	29.29/.7534	29.22/.7533	29.27/.7522	**29.37/.7560**
	50	27.19/.6747	27.04/.6616	27.08/.6600	27.34/.6767	27.44/**.6831**	27.32/.6814	27.30/.6733	**27.48**/.6830
	70	25.93/.6226	25.86/.6184	25.83/.6063	26.14/.6296	**26.22/.6320**	26.11/.6347	26.02/.6232	**26.22**/.6289
House	10	36.71/.9218	36.66/.9187	36.68/.9269	36.95/.9236	36.48/.9108	36.93/.9263	36.97/.9241	**37.02/.9276**
	30	32.09/.8480	32.30/.8508	32.12/.8475	32.52/.8514	32.24/.8498	32.41/.8524	32.50/.8517	**32.59/.8548**
	50	29.69/.8122	30.17/.8240	29.41/.7983	30.32/.8222	30.00/.8180	30.13/.8241	30.32/.8163	**30.56/.8305**
	70	27.91/.7747	28.41/.7960	27.51/.7536	28.60/.7957	28.19/.7805	28.36/.7958	28.68/.7996	**28.86/.8077**
Lena	10	35.93/.9166	35.90/.9168	35.93/.9179	36.06/.9180	**36.18/.9188**	36.02/.9170	36.07/.9183	36.03/.9176
	30	31.26/.8449	31.27/.8486	31.49/.8525	31.43/.8498	**31.59**/.8543	31.46/.8513	31.48/.8515	31.53/**.8552**
	50	29.05/.7994	29.01/.8046	29.20/.8026	29.25/.8053	**29.39**/.8116	29.27/.8090	29.14/.7966	29.37/**.8123**
	70	27.57/.7603	27.55/.7729	27.62/.7586	27.85/.7739	27.85/.7701	27.80/.7761	27.76/.7727	**28.04/.7849**
Man	10	33.98/.9076	34.12/.9092	34.13/.9109	34.23/.9113	**34.44/.9154**	34.14/.9104	34.26/.9118	34.28/.9127
	30	28.86/.7802	28.81/.7741	28.92/.7752	29.00/.7830	**29.30/.7953**	28.94/.7837	29.04/.7842	29.13/.7887
	50	26.81/.7056	26.68/.6977	26.77/.6946	26.94/.7091	**27.24/.7217**	26.87/.7101	26.94/.7052	27.10/.7160
	70	25.56/.6548	25.42/.6526	25.44/.6396	25.68/.6604	**25.91/.6681**	25.59/.6622	25.63/.6573	25.82/.6660
Montage	10	37.35/.9679	37.46/.9681	37.73/.9690	37.84/.9692	37.60/.9690	37.61/.9695	37.83/.9688	**37.93/.9697**
	30	31.38/.9114	31.06/.9195	**32.07**/.9198	31.65/.9183	31.87/.9235	31.57/.9209	31.47/.9192	31.97/**.9271**
	50	27.90/.8614	28.00/.8763	28.73/.8671	28.27/.8741	28.98/.8814	28.07/.8751	28.37/.8679	**29.11/.8874**
	70	25.92/.8116	25.84/.8300	26.40/.8148	26.11/.8303	26.69/.8333	26.03/.8304	26.13/.8370	**27.08/.8583**
Peppers	10	34.68/.9282	34.82/.9288	34.76/.9290	34.95/.9300	**35.12/.9318**	34.80/.9286	34.99/.9305	34.96/.9304
	30	29.28/.8505	29.24/.8536	29.54/.8552	29.49/.8557	**29.91/.8647**	29.42/.8562	29.49/.8566	29.58/.8601
	50	26.68/.7936	26.73/.7993	26.92/.7942	26.91/.7999	**27.32/.8106**	26.81/.7969	26.99/.7995	27.13/.8088
	70	25.07/.7477	24.97/.7479	25.07/.7389	25.26/.7525	25.48/.7598	25.23/.7473	25.34/.7617	**25.53/.7714**
Average	10	34.71/.9241	34.79/.9238	34.68/.9250	34.98/.9262	34.91/.9261	34.88/.9257	**35.00**/.9266	34.99/**.9271**
	30	29.57/.8327	29.54/.8305	29.67/.8313	29.79/.8366	29.78/.8392	29.74/.8381	29.78/.8376	**29.91/.8422**
	50	27.13/.7697	27.16/.7687	27.21/.7631	27.41/.7771	27.44/.7788	27.34/.7796	27.38/.7714	**27.61/.7845**
	70	25.07/.7477	25.62/.7240	25.57/.7078	25.88/.7316	25.74/.7223	25.82/.7340	25.81/.7289	**26.09/.7398**

[15]. This experiment demonstrates that both stages of the proposed method are necessary and generally lead to significant improvements in PSNR/SSIM and visual quality.

Next, we comprehensively compared the KWFM performance with the state-of-the-art methods, such as BM3D[3] [7], SAIST [9], DDID2[4] [22], WNNM[5] [15] and DnCNN-S[6] [36]. The experimental results of all the benchmark methods were produced using the source codes or executable programs with default parameters released by the authors [7,9,15,22,36]. The calculated image quality metrics are also given in Table 2 to Table 3, which show that DnCNN-S [36] achieves the best averaged PSNR/SSIM results and KWFM is the second best for one dataset 'BSD68'. Furthermore, KWFM is the best for the other dataset 'Set11'.

Fig. 4 and Fig. 5 give visual comparisons of denoised images for two test grayscale images from the dataset 'BSD68'. Fig. 6, Fig. 7 and Fig. 8 give visual comparisons of denoised images for test grayscale images *'Boats'*, *'Hill'* and *'House'* from the

[3] http://www.cs.tut.fi/~foi/GCF-BM3D/BM3D.zip.
[4] http://www.cgg.unibe.ch/publications/dual-domain-filtering/ddf_code.zip.
[5] http://www4.comp.polyu.edu.hk/~cslzhang/code/WNNM_code.zip.
[6] https://github.com/cszn/DnCNN.

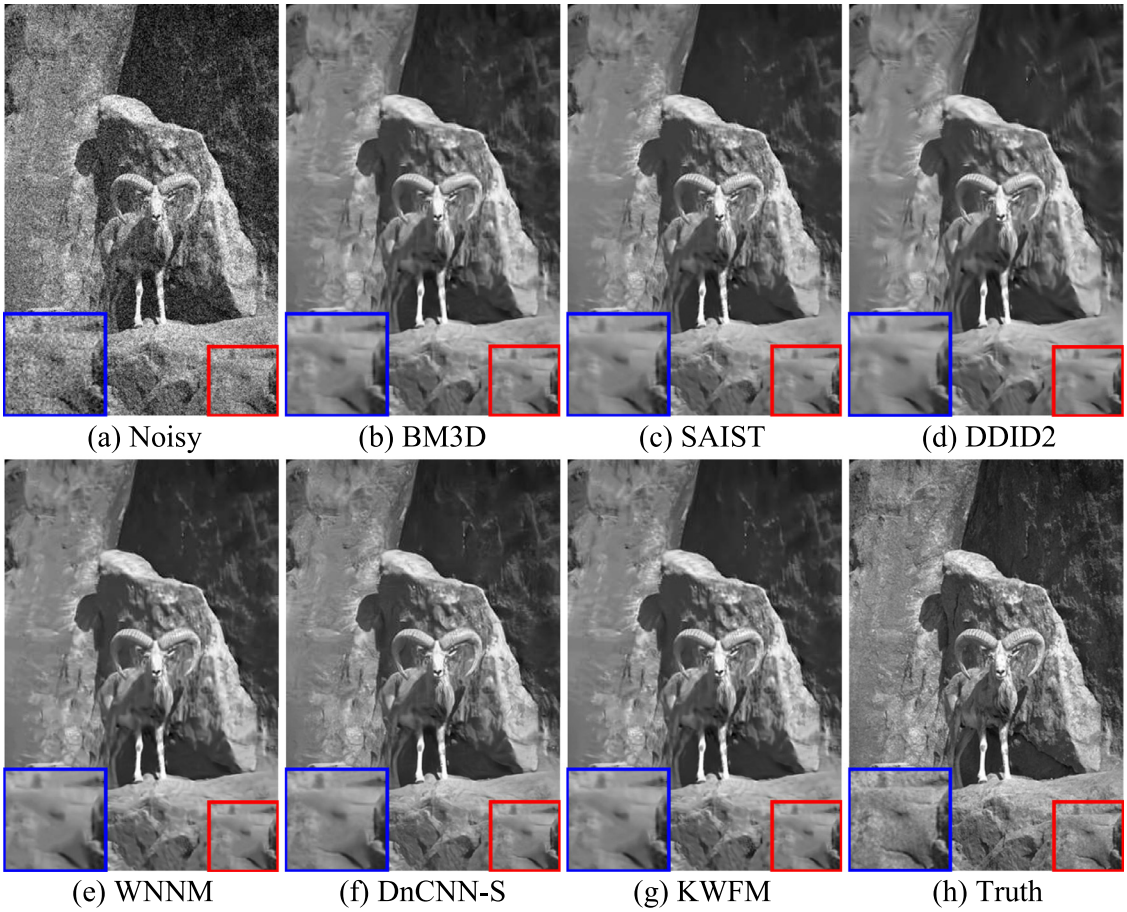


Fig. 4. Visual comparison of the denoised results obtained by BM3D [7], SAIST [9], DDID2 [22], WNNM [15], DnCNN-S [36] and our KWFM for one example from the grayscale image dataset 'BSD68' corrupted by the Gaussian noise with standard deviation 30. As test images are consistent with the patterns of training images, DnCNN-S [36] recovers more detailed structures and has better denoising results than our KWFM for one grayscale test image from the dataset 'BSD68'. Our KWFM is the second best and approaches to deep learning based methods including DnCNN-S [36].

Table 4
Averaged PSNR (dB) and SSIM results of these different methods for the color image dataset 'CBSD68' with variant noise levels. The best results are highlighted in boldface.

Noise level	BM3D [7]	DDID2 [22]	DnCNN-S [36]	KWFM
10	35.90/.9510	35.43/.9499	**36.36/.9559**	35.98/.9531
30	29.72/.8421	29.88/.8442	**30.41/.8645**	30.07/.8548
50	27.37/.7624	27.54/.7604	**27.97/.7928**	27.60/.7735
70	26.00/.7069	26.07/.6943	**26.55/.7404**	26.17/.7174

dataset 'Set11'. Fig. 4 to Fig. 5 shows that DnCNN-S [36] recovers more detailed structures and has better denoising results than BM3D [7], SAIST [9], DDID2 [22], WNNM [15] and our KWFM for the dataset 'BSD68'. However, Fig. 6 to Fig. 8 shows that KWFM generally achieves better visual results with less artifacts than the competing methods for test grayscale images from the dataset 'Set11'.

3.4. Results for color images

For the color images, we adopted the color image datasets 'CBSD68' and 'Set12' to compare KWFM with the state-of-the-art methods, such as BM3D [7], DDID2 [22], DnCNN-S [36]. In this experiment, the results of the existing methods were produced with the default color space conversion provided by the authors [7,22,36]. Table 4 and 5 show the PSNR/SSIM results of these methods for 'CBSD68' and 'Set12', respectively. Here both PSNR (dB) and SSIM were computed as the mean score from the three channels between the denoised RGB image and the ground-truth RGB image. From Table 4 to Table 5, DnCNN-S [36] has the highest averaged PSNR/SSIM and our KWFM is the second best for one dataset 'CBSD68', but our

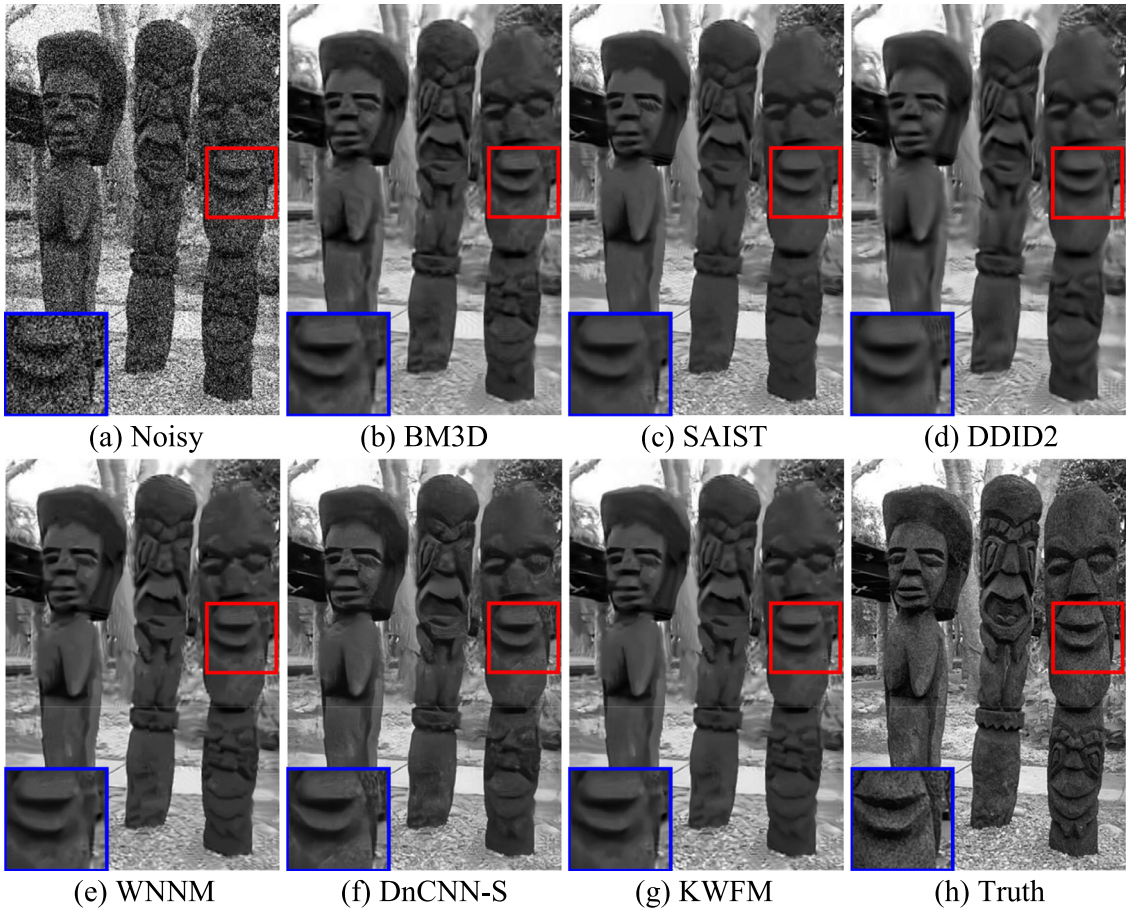


Fig. 5. Visual comparison of the denoised results obtained by BM3D [7], SAIST [9], DDID2 [22], WNNM [15], DnCNN-S [36] and our KWFM for another example from the grayscale image dataset 'BSD68' corrupted by the Gaussian noise with standard deviation 50. As test images are consistent with the patterns of training images, DnCNN-S [36] recovers more detailed structures and has better denoising results than our KWFM for another grayscale test image from the dataset 'BSD68'. Our KWFM is the second best and approaches to deep learning based methods including DnCNN-S [36].

KWFM achieves better results than the competing methods for the other dataset 'Set12'. Besides the quantitative evaluation, subjective visual assessments were carried out to further inspect the effectiveness of our KWFM. Fig. 9 and 10 show the denoising results for two color images from the dataset 'CBSD68' and two color images *'Tiffany'* and *'House'* from the dataset 'Set12', respectively. Fig. 9 shows that DnCNN-S [36] has the best visual results for the dataset 'CBSD68'. However, Fig. 10 shows that our KWFM achieves better visual results with more details than BM3D [7], DDID2 [22] and DnCNN-S [36] for the dataset 'Set12'.

The above quantitative and qualitative evaluations demonstrate that our KWFM can not only remove the noise effectively but also recover fine structures and sharp edges. Moreover, our KWFM can generally achieve less visual artifacts and color distortion than the leading state-of-the-art methods, except for DnCNN-S [36] in certain cases. The code for our KWFM will be provided on the website[7] in the spirit of reproducible research.

3.5. Computational complexity

We further analyzed the computational complexity of KWFM, which was implemented in Matlab 8.2 running in Windows 10 operating system on ThinkPad S1 Yoga with 4 Intel(R) Core(TM) i7-4510U CPUs @2.00GHz 8.00GB RAM. For the evaluation of the computational complexity, we tested the baseline methods and our KWFM on the grayscale images (e.g., *'C.man'* and *'Lena'*) with noise deviation 30. Note that the source code of BM3D [7] was written in the C programming language, whereas the source programs of SAIST [9], DDID2 [22], WNNM [15] and DnCNN-S [36] were in Matlab. Table 6 shows the comparison of computation times between these denoising methods, which shows that our KWFM algorithm without code optimization

[7] https://github.com/zhangyongqin/KWFM.

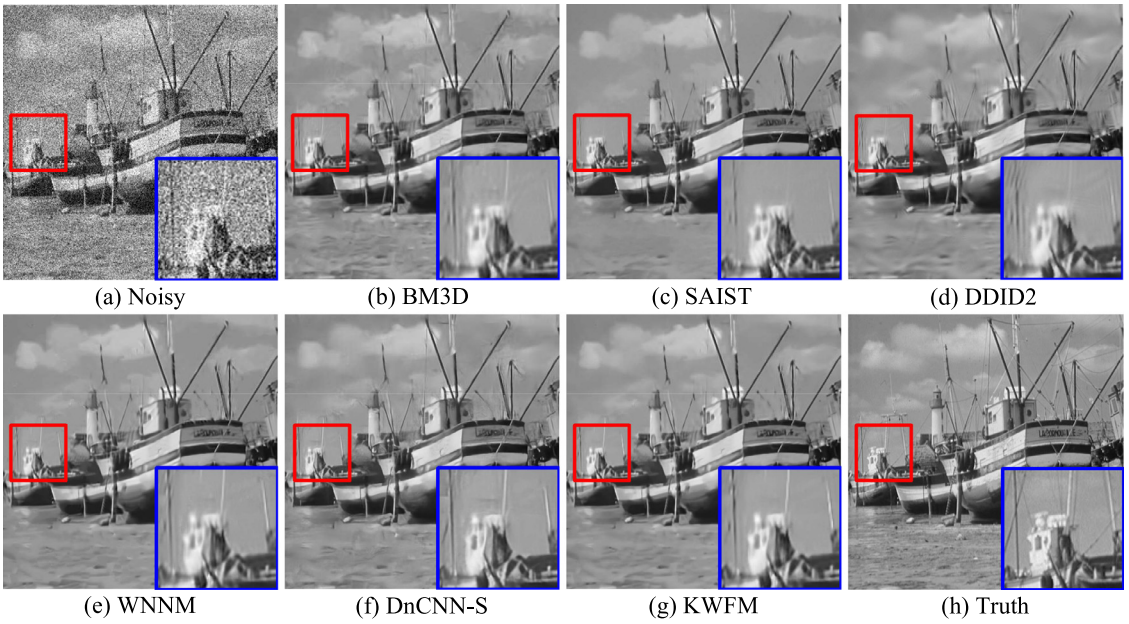


Fig. 6. Visual comparison of the denoised results obtained by BM3D [7], SAIST [9], DDID2 [22], WNNM [15], DnCNN-S [36] and our KWFM for the grayscale image *'Boats'* corrupted by the Gaussian noise with standard deviation 50. As there are discrepancies between test images and training images, our KWFM recovers more detailed structures and has better denoising results than competing methods, even DnCNN-S [36], for one grayscale test image from the dataset 'Set11'.

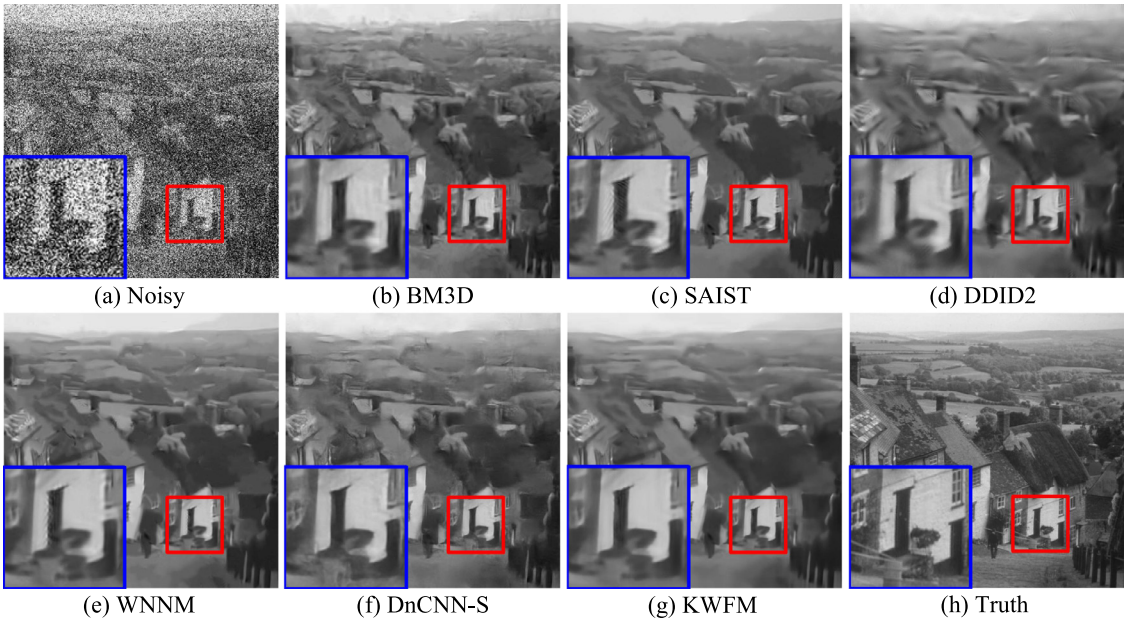


Fig. 7. Visual comparison of the denoised results obtained by BM3D [7], SAIST [9], DDID2 [22], WNNM [15], DnCNN-S [36] and our KWFM for the grayscale image *'Hill'* corrupted by the Gaussian noise with standard deviation 70. As there are discrepancies between test images and training images, our KWFM recovers more detailed structures and has better denoising results than competing methods, even DnCNN-S [36], for another grayscale test image from the dataset 'Set11'.

takes longer than BM3D [7], SAIST [9] and DnCNN-S [36], but is faster than DDID2 [22] and WNNM [15]. Similar results were obtained for other images and noise levels.

We have also carried out theoretical analysis of the computational complexity of the proposed method. Suppose that there are n patches of size $p \times p$ for an input $M \times N$ image. The computational complexity of the proposed method mainly consists of two parts: OLRA and KWF. In the first part, the block-matching search needs $\mathcal{O}\left(nW^2\left(p^2+Q\right)\right)$ op-

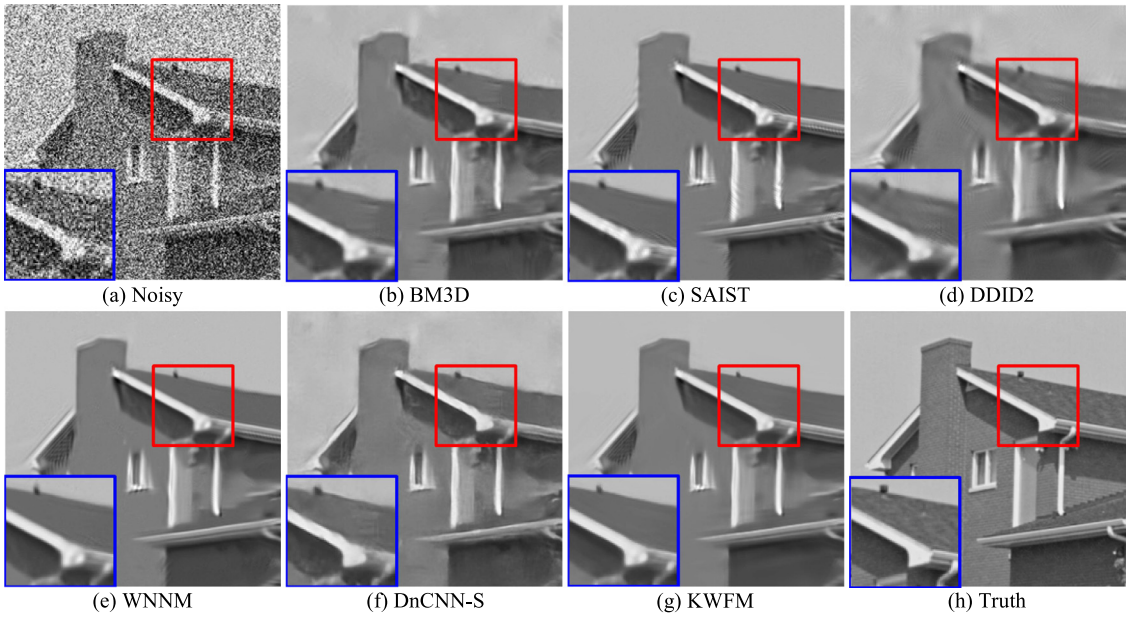


Fig. 8. Visual comparison of the denoised results obtained by BM3D [7], SAIST [9], DDID2 [22], WNNM [15], DnCNN-S [36] and our KWFM for the grayscale image *'House'* corrupted by the Gaussian noise with standard deviation 70. As there are discrepancies between test images and training images, our KWFM recovers more detailed structures and has better denoising results than competing methods, even DnCNN-S [36], for another grayscale test image from the dataset 'Set11'.

Table 5
PSNR (dB) and SSIM results of BM3D [7], DDID2 [22], DnCNN-S [36] and our KWFM separately applied to the color image dataset 'Set12' with variant noise levels. The best results are highlighted in boldface.

Images	10				30				50				70			
	[7]	[22]	[36]	KWFM	[7]	[22]	[36]	KWFM	[7]	[22]	[36]	KWFM	[7]	[22]	[36]	KWFM
Baboon	30.64	30.66	28.63	**30.78**	25.14	**25.50**	25.09	25.45	23.15	**23.46**	23.30	23.43	21.97	22.16	**22.20**	22.19
	.9105	**.9178**	.8695	.9164	.7568	.7719	.7545	**.7751**	.6437	.6629	.6662	**.6799**	.5585	.5683	.5976	**.6024**
F16	36.68	36.59	36.41	**36.88**	31.93	32.31	31.44	**32.41**	29.79	30.04	29.19	**30.27**	28.28	28.38	27.76	**28.83**
	.9359	.9354	.9349	**.9370**	.8825	.8847	.8875	**.8897**	.8490	.8429	.8531	**.8547**	.8193	.8026	.8252	**.8324**
House	36.23	35.83	34.98	**36.38**	32.34	32.26	31.10	**32.65**	30.47	30.36	29.05	**31.05**	29.02	28.79	27.85	**29.94**
	.9107	.9104	.8943	**.9164**	.8336	.8334	.8284	**.8399**	.8015	.7930	.8031	**.8086**	.7758	.7563	.7829	**.7956**
Lake	32.31	32.34	30.99	**32.60**	28.02	**28.37**	27.88	28.32	26.28	26.59	26.21	**26.62**	25.13	25.32	25.08	**25.45**
	.8666	.8739	.8311	**.8849**	.7624	.7660	.7616	**.7665**	.7124	.7120	.7167	**.7201**	.6760	.6674	.6819	**.6868**
Lena	35.22	35.19	34.66	**35.27**	31.59	**31.86**	31.54	31.84	29.88	30.04	29.70	**30.12**	28.63	28.66	28.50	**28.96**
	.8863	.8877	.8754	**.8878**	.8241	**.8291**	.8236	.8290	.7894	.7887	.7875	**.7940**	.7591	.7516	.7605	**.7702**
Peppers	33.78	33.87	32.99	**33.94**	30.61	**30.90**	30.18	30.72	28.93	**29.31**	28.36	29.21	27.66	28.02	27.27	**28.12**
	.8408	.8470	.8179	**.8501**	.7659	**.7732**	.7667	.7650	.7320	**.7364**	.7307	.7356	.7040	.7040	.7046	**.7159**
Splash	37.56	37.58	37.08	**37.70**	34.30	34.48	33.26	**34.68**	32.37	32.65	31.30	**33.19**	30.79	31.08	30.08	**32.03**
	.9040	**.9051**	.9027	.9049	.8591	.8616	.8551	**.8627**	.8366	.8333	.8298	**.8461**	.8150	.8036	.8125	**.8361**
Tiffany	35.49	**35.84**	34.43	35.61	31.56	**32.00**	30.79	31.93	29.83	30.05	28.99	**30.23**	28.53	28.73	27.90	**29.14**
	.8928	**.9004**	.8806	.8934	.8199	**.8280**	.8200	.8277	.7839	.7821	.7816	**.7901**	.7560	.7445	.7561	**.7692**
Kodim01	34.71	34.43	**34.99**	34.90	28.14	28.36	**28.80**	28.72	25.86	25.94	**26.47**	26.39	24.60	24.43	**25.14**	25.06
	.9517	.9515	**.9551**	.9548	.8087	.8192	**.8402**	.8353	.7014	.7035	**.7495**	.7363	.6307	.6101	**.6841**	.6617
Kodim02	36.56	36.63	**36.98**	36.97	31.72	31.76	**32.18**	32.08	29.84	29.82	**30.23**	30.21	28.62	28.59	29.05	**29.20**
	.9133	.9184	.9229	**.9241**	.8021	.7983	**.8196**	.8125	.7471	.7357	**.7617**	.7503	.7131	.6939	**.7270**	.7189
Kodim03	39.05	38.87	39.23	**39.27**	33.58	33.85	34.00	**34.13**	31.34	31.32	31.55	**31.79**	30.00	29.72	30.21	**30.34**
	.9564	.9564	.9572	**.9578**	.8911	.8953	.9011	**.9034**	.8440	.8349	.8536	**.8557**	.8085	.7849	**.8203**	.8201
Kodim12	37.84	37.69	**38.14**	38.09	32.96	32.87	**33.33**	33.27	30.98	30.80	31.22	**31.22**	29.78	29.50	30.00	**30.02**
	.9323	.9332	**.9376**	.9368	.8471	.8390	**.8566**	.8512	.7956	.7801	**.8030**	.7925	.7628	.7394	**.7717**	.7611
Average	35.51	35.46	34.96	**35.70**	30.99	31.21	30.80	**31.35**	29.06	29.20	28.80	**29.48**	27.75	27.78	27.59	**28.27**
	.9084	.9115	.8983	**.9137**	.8211	.8250	.8262	**.8298**	.7697	.7671	.7780	**.7803**	.7316	.7189	.7437	**.7475**

Table 6
Comparison of computational complexity (unit: seconds).

Size	BM3D [7]	SAIST [9]	DDID2 [22]	WNNM [15]	DnCNN-S [36]	KWFM
256×256	1.48	40.42	302.69	438.80	31.91	201.58
512×512	4.24	146.08	1166.61	1807.29	138.01	855.35

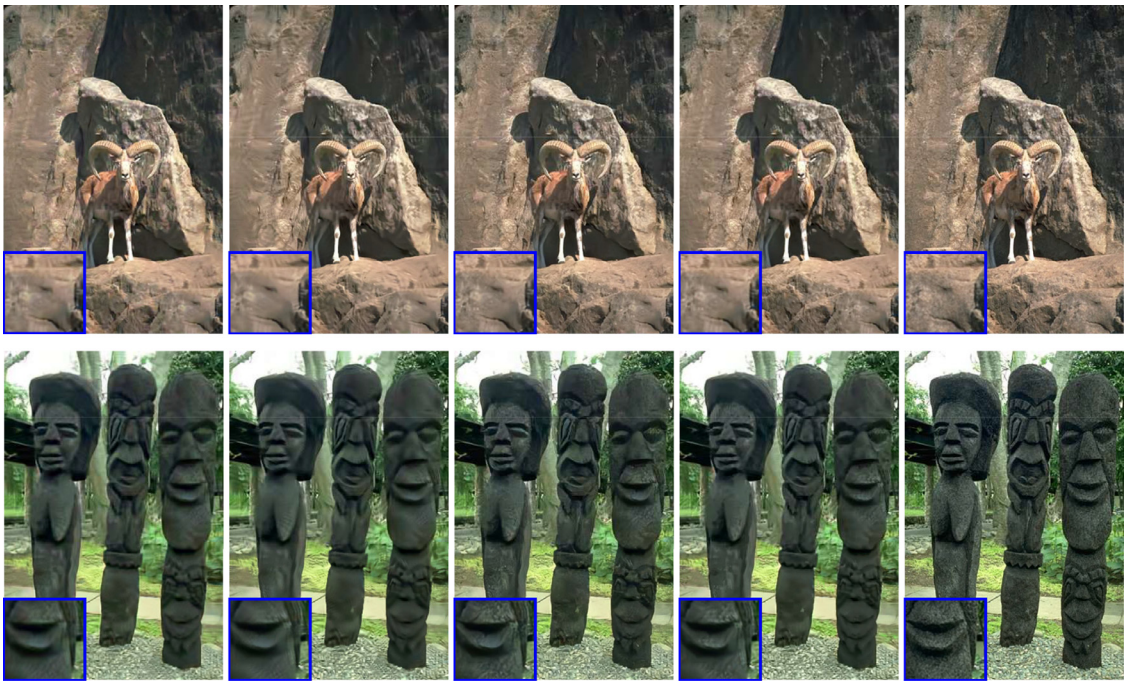


Fig. 9. Visual comparison of the denoised results obtained by BM3D [7], DDID2 [22], DnCNN-S [36] and our KWFM for two test examples from the color image database 'CBSD68'. These two images are separately corrupted by the Gaussian noise with standard deviation 30 and 50. As test images are consistent with the patterns of training images, DnCNN-S [36] recovers more detailed structures and has better denoising results than our KWFM for color test images from the dataset 'CBSD68'. Our KWFM is the second best and approaches to deep learning based methods including DnCNN-S [36].

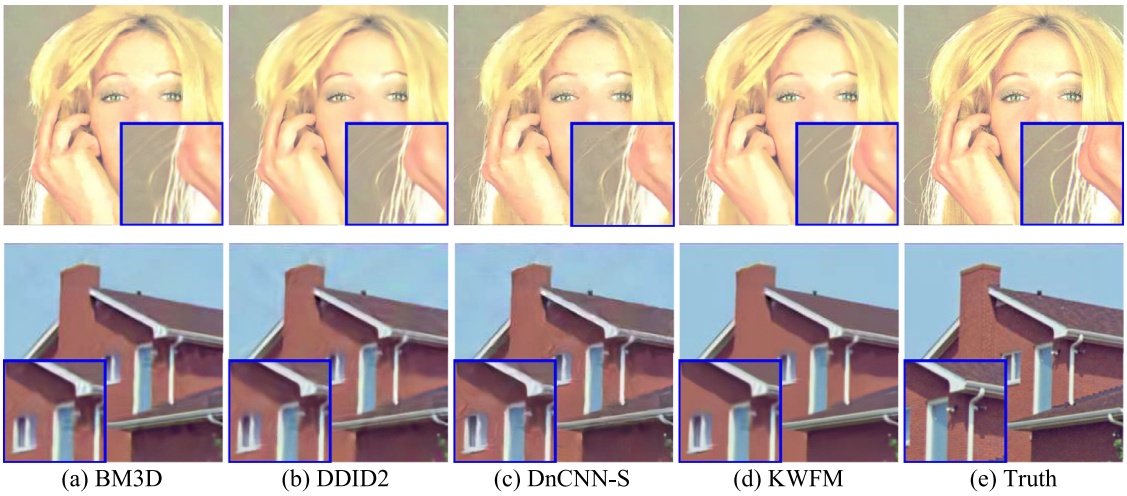


Fig. 10. Visual comparison of the denoised results obtained by BM3D [7], DDID2 [22], DnCNN-S [36] and our KWFM for test color images *'Tiffany'* and *'House'*. These two images are separately corrupted by the Gaussian noise with standard deviation 50 and 70. As there are discrepancies between test images and training images, our KWFM recovers more detailed structures and has better denoising results than competing methods, even DnCNN-S [36], for color test images from the dataset 'Set12'.

erations, while the reference image update needs $\mathcal{O}\left(n\left(p^4Q+Q^3\right)\right)$. Therefore, the number of operations of OLRA is $\mathcal{O}\left(nT\left(p^4Q+Q^3+p^2W^2+W^2Q\right)\right)$, where T is the number of iterations. In the second part, KWF approximately costs $\mathcal{O}\left(nL^2\log L^2\right)$. Therefore, the total computational complexity of the proposed method is

$$\mathcal{O}\left(nT\left(p^4Q+Q^3+p^2W^2+W^2Q\right)+nL^2\log L^2\right). \tag{16}$$

4. Discussion

To recover complex image structures, we propose the kernel Wiener filtering model for image denoising by introducing the shape-aware kernel function, and estimate the reference image of kernel Wiener filtering by OLRA. Our model consists of two subproblems, where we deduce a closed-form solution of the optimal KWF and a convergent iterative solution of nearly unbiased low-rank estimator for the reference images, which distinguish our KWFM from the existing methods [4,7,9,21,22]. In fact, the first stage of our KWFM is an improved version of SVT [4,9], and usually achieves higher estimation accuracy than SVT [4,9] for the low-rank approximation. The second stage is a generalized form of CWF in BM3D [7]. Essentially, CWF is a special case of our KWF when the elements of the kernel function are all ones.

DnCNN-S [36] often achieves better results than our KWFM for test images in the dataset 'BSD68' or its color version because it uses the remaining 432 images of Berkeley Segmentation Dataset 'BSDS500'[8] as the training images. This implies that DnCNN-S [36] can produce excellent results when the test images are consistent with the patterns of the training images. On the other hand, our KWFM generally has better results than DnCNN-S [36] for test images from the datasets 'Set11' and 'Set12' because these test images deviate from the patterns of the training images used by DnCNN-S [36]. Considering the balance between the denoising performance and the computational complexity, our KWFM is effective and robust for noise reduction and often superior to the current state-of-the-art methods including DnCNN-S [36] in certain cases.

5. Conclusions

In this paper, we have presented a novel kernel Wiener filtering model for image denoising. As an extended version of conventional Wiener filtering, a kernel Wiener filtering method is designed by introducing a shape-aware kernel function, where its reference image is estimated by an optimized low-rank approximation approach. By breaking this model into two subproblems, we separately derive an optimal kernel Wiener filter in a closed-form solution and a nearly unbiased low-rank estimate of the reference images in a convergent iterative solution. This optimized low-rank estimation based on eigenvalue thresholding achieves higher estimation accuracy than conventional low-rank approximation based on singular value thresholding. Experimental results demonstrated that the proposed algorithm can faithfully recover image details with good robustness while removing noise effectively, and generally outperform the current state-of-the-art methods both visually and quantitatively. In addition, due to the simplicity and superior performance of kernel Wiener filtering, our method can be easily combined with existing denoising methods for further improving the performance.

Acknowledgements

This work was supported by Scientific Research Program Funded by Shaanxi Provincial Education Department (Program No. 16JK1762), Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2016JQ6068), Program for Changjiang Scholars and Innovative Research Team in University (Grant No. IRT13090), and National Natural Science Foundation of China (Grant Nos. 61471272 and 61201442).

Appendix A. Eigenvalues of Covariance Matrix

According to the noisy image observation $\mathbf{G} = \mathbf{O} + \varsigma$, we deduce the relationship between the eigenvalues of the covariance matrix for the noisy observation $\mathbf{G} \in \mathbb{R}^{p^2 \times Q}$ and those for its latent clean image $\mathbf{O} \in \mathbb{R}^{p^2 \times Q}$. For convenience, let $\Sigma_{\mathbf{G}}$ and $\Sigma_{\mathbf{O}}$ denote diagonal matrices form by the eigenvalues of $\mathbf{G}$ and $\mathbf{O}$, respectively. ς is assumed to follow a Gaussian distribution with zero mean and variance σ_ς^2. If the noise ς is independently identically distributed (i.i.d.), then $\Sigma_{\mathbf{G}}^2 \simeq \Sigma_{\mathbf{O}}^2 + Q\sigma_\varsigma^2$.

Proof. The eigenvalue decomposition of the covariance matrix $\mathbf{G}\mathbf{G}'$ can be expressed as

$$\begin{aligned} \mathbf{G}\mathbf{G}' &= \mathbf{U}\Sigma_{\mathbf{G}}^2\mathbf{U}' \simeq \mathbf{O}\mathbf{O}' + \varsigma\varsigma' \\ &\simeq \mathbf{U}\Sigma_{\mathbf{O}}^2\mathbf{U}' + Q\sigma_\varsigma^2\mathbf{I} = \mathbf{U}\left(\Sigma_{\mathbf{O}}^2 + Q\sigma_\varsigma^2\right)\mathbf{U}', \end{aligned} \tag{A.1}$$

where $\mathbf{U}$ denotes the eigenvectors, and $\mathbf{I}$ is an identity matrix. This completes the proof. □

References

[1] T.F. Andre, R.D. Nowak, B.D. Van Veen, Low-rank estimation of higher order statistics, IEEE Trans. Signal Processi. 45 (3) (1997) 673–685.
[2] A. Bloemendal, A. Knowles, H.T. Yau, J. Yin, On the principal components of sample covariance matrices, Probab. Theory Related Fields 164 (1-2) (2016) 459–552.
[3] A. Buades, B. Coll, J.M. Morel, A review of image denoising algorithms, with a new one, SIAM Multiscale Model. Simul. 4 (2) (2005) 490–530.
[4] J. Cai, E. Candes, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim. 20 (4) (2010) 1956–1982.
[5] Y. Chen, Y. Guo, Y. Wang, D. Wang, C. Peng, G. He, Denoising of hyperspectral images using nonconvex low rank matrix approximation, IEEE Trans.Geosci.Remote Sens. 55 (9) (2017) 5366–5380.

[8] Berkeley Segmentation Data Set and Benchmarks 500 (BSDS500), March, 2013, https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html.

[6] Y. Chen, T. Pock, Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2017) 1256–1272.
[7] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-D transform-domain collaborative filtering, IEEE Trans. Image Process. 16 (8) (2007) 2080–2095.
[8] N. Divakar, R.V. Babu, Image denoising via cnns: An adversarial approach, in: In New Trends in Image Restoration and Enhancement, CVPR workshop, 2017, pp. 1076–1083.
[9] W. Dong, G. Shi, X. Li, Nonlocal image restoration with bilateral variance estimation: A low-rank approach, IEEE Trans. Image Process. 22 (2) (2013) 700–711.
[10] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Trans. Image Process. 15 (12) (2006) 3736–3745.
[11] F. Fan, Y. Ma, C. Li, X. Mei, J. Huang, J. Ma, Hyperspectral image denoising with superpixel segmentation and low-rank representation, Inf. Sci. 397 (2017) 48–68.
[12] V. Fedorov, C. Ballester, Affine non-local means image denoising, IEEE Trans. Image Process. 26 (5) (2017) 2137–2148.
[13] S. Gai, Multiresolution monogenic wavelet transform combined with bivariate shrinkage functions for color image denoising, Circuits Syst. Signal Process. 37 (3) (2018) 1162–1176.
[14] M. Georgiev, R. Bregovic, A. Gotchev, Time-of-flight range measurement in low-sensing environment: Noise analysis and complex-domain non-local denoising, IEEE Trans. Image Process. 27 (6) (2018) 2911–2926.
[15] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, L. Zhang, Weighted nuclear norm minimization and its applications to low level vision, Int. J. Comput. Vision 121 (2) (2017) 183–208.
[16] Y.M. Huang, H.Y. Yan, Y.W. Wen, X. Yang, Rank minimization with applications to image noise removal, Inf. Sci. 429 (2018) 147–163.
[17] A.N. Iusem, On the convergence properties of the projected gradient method for convex optimization, Comput. Appl. Math. 22 (1) (2003) 37–52.
[18] P. Jain, V. Tyagi, LAPB: Locally adaptive patch-based wavelet domain edge-preserving image denoising, Inf. Sci. 294 (2015) 164–181.
[19] Q. Jin, I. Grama, C. Kervrann, Q. Liu, Non-local means and optimal weights for noise removal, SIAM J. Imaging Sci. 10 (4) (2017) 1878–1920.
[20] A. Khmag, A. Ramli, S. Al-haddad, S. Yusoff, N. Kamarudin, Denoising of natural images through robust wavelet thresholding and genetic programming, Visual Comput. 33 (9) (2017) 1141–1154.
[21] C. Knaus, M. Zwicker, Progressive image denoising, IEEE Trans. Image Process. 23 (7) (2014) 3114–3125.
[22] C. Knaus, M. Zwicker, Dual-domain filtering, SIAM J. Imaging Sci. 8 (3) (2015) 1396–1420.
[23] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[24] S. Lefkimmiatis, Non-local color image denoising with convolutional neural networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5882–5891.
[25] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proceedings of IEEE International Conference on Computer Vision, 2, 2001, pp. 416–423.
[26] G. Papari, N. Idowu, T. Varslot, Fast bilateral filtering for denoising large 3D images, IEEE Trans. Image Process. 26 (1) (2017) 251–261.
[27] J. Portilla, V. Strela, M.J. Wainwright, E.P. Simoncelli, Image denoising using scale mixtures of Gaussians in the wavelet domain, IEEE Trans. Image Process. 12 (11) (2003) 1338–1351.
[28] L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D 60 (1-4) (1992) 259–268.
[29] C. Tomasi, R. Manduchi, Bilateral filtering for gray and color images, in: Proceedings of IEEE International Conference on Computer Vision, 1998, pp. 839–846.
[30] O. Tuzel, F. Porikli, P. Meer, Region covariance: A fast descriptor for detection and classification, in: Proceedings of European Conference on Computer Vision, Part II, 2006, pp. 589–600.
[31] J.H. Wang, F.Y. Meng, L.P. Pang, X.H. Hao, An adaptive fixed-point proximity algorithm for solving total variation denoising models, Inf. Sci. 402 (2017) 69–81.
[32] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[33] J. Xiao, H. Tian, Y. Zhang, Y. Zhou, J. Lei, Blind video denoising via texture-aware noise estimation, Comput. Vision Image Understanding 169 (2018) 1–13.
[34] J. Xue, Y. Zhao, W. Liao, S.G. Kong, Joint spatial and spectral low-rank regularization for hyperspectral image denoising, IEEE Trans. Geosci. Remote Sensing 56 (4) (2018) 1940–1958.
[35] H. Yoshino, C. Dong, Y. Washizawa, Y. Yamashita, Kernel Wiener filter and its application to pattern recognition, IEEE Trans. Neural Networks 21 (11) (2010) 1719–1730.
[36] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising, IEEE Trans. Image Process. 26 (7) (2017) 3142–3155.
[37] Y. Zhang, J. Liu, M. Li, Z. Guo, Joint image denoising using adaptive principal component analysis and self-similarity, Inf. Sci. 259 (2014) 128–141.
[38] Y. Zhang, J. Liu, W. Yang, Z. Guo, Image super-resolution based on structure-modulated sparse representation, IEEE Trans. Image Process. 24 (9) (2015) 2797–2810.
[39] Y. Zhang, F. Shi, J. Cheng, L. Wang, P.T. Yap, D. Shen, Longitudinally guided super-resolution of neonatal brain magnetic resonance images, IEEE Trans. Cybernetics (2018), doi:10.1109/TCYB.2017.2786161.
[40] W. Zhao, H. Lu, Medical image fusion and denoising with alternating sequential filter and adaptive fractional order total variation, IEEE Trans. Instrumen. Measure. 66 (9) (2017) 2283–2294.