A Trilateral Weighted Sparse Coding Scheme for Real-World Image Denoising

Jun Xu[1], Lei Zhang[1(✉)], and David Zhang[1,2]

[1] The Hong Kong Polytechnic University, Kowloon, Hong Kong SAR, China
{csjunxu,cslzhang,csdzhang}@comp.polyu.edu.hk
[2] School of Science and Engineering, The Chinese University of Hong Kong
(Shenzhen), Shenzhen, China

Abstract. Most of existing image denoising methods assume the corrupted noise to be additive white Gaussian noise (AWGN). However, the realistic noise in real-world noisy images is much more complex than AWGN, and is hard to be modeled by simple analytical distributions. As a result, many state-of-the-art denoising methods in literature become much less effective when applied to real-world noisy images captured by CCD or CMOS cameras. In this paper, we develop a trilateral weighted sparse coding (TWSC) scheme for robust real-world image denoising. Specifically, we introduce three weight matrices into the data and regularization terms of the sparse coding framework to characterize the statistics of realistic noise and image priors. TWSC can be reformulated as a linear equality-constrained problem and can be solved by the alternating direction method of multipliers. The existence and uniqueness of the solution and convergence of the proposed algorithm are analyzed. Extensive experiments demonstrate that the proposed TWSC scheme outperforms state-of-the-art denoising methods on removing realistic noise.

Keywords: Real-world image denoising · Sparse coding

1 Introduction

Noise will be inevitably introduced in imaging systems and may severely damage the quality of acquired images. Removing noise from the acquired image is an essential step in photography and various computer vision tasks such as segmentation [1], HDR imaging [2], and recognition [3], etc. Image denoising aims to recover the clean image $\mathbf{x}$ from its noisy observation $\mathbf{y} = \mathbf{x} + \mathbf{n}$, where $\mathbf{n}$ is the corrupted noise. This problem has been extensively studied in literature, and numerous statistical image modeling and learning methods have been proposed in the past decades [4–26].

Most of the existing methods [4–20] focus on additive white Gaussian noise (AWGN), and they can be categorized into dictionary learning based methods [4,5], nonlocal self-similarity based methods [6–14], sparsity based methods [4,5,7–11], low-rankness based methods [12,13], generative learning based methods [14–16], and discriminative learning based methods [17–20], etc. However, the realistic noise in real-world images captured by CCD or CMOS cameras is much more complex than AWGN [21–24,26–28], which can be signal dependent and vary with different cameras and camera settings (such as ISO, shutter speed, and aperture, etc.). In Fig. 1, we show a real-world noisy image from the Darmstadt Noise Dataset (DND) [29] and a synthetic AWGN image from the Kodak PhotoCD Dataset (http://r0k.us/graphics/kodak/). We can see that the different local patches in real-world noisy image show different noise statistics, e.g., the patches in black and blue boxes show different noise levels although they are from the same white object. In contrast, all the patches from the synthetic AWGN image show homogeneous noise patterns. Besides, the realistic noise varies in different channels as well as different local patches [22–24,26]. In Fig. 2, we show a real-world noisy image captured by a Nikon D800 camera with ISO = 6400, its "Ground Truth" (please refer to Sect. 4.3), and their differences in full color image as well as in each channel. The overall noise standard deviations (stds) in Red, Green, and Blue channels are 5.8, 4.4, and 5.5, respectively. Besides, the realistic noise is inhomogeneous. For example, the stds of noise in the three boxes plotted in Fig. 2(c) vary largely. Indeed, the noise in real-world noisy image is much more complex than AWGN noise. Though having shown promising performance on AWGN noise removal, many of the above mentioned methods [4–20] will become much less effective when dealing with the complex realistic noise as shown in Fig. 2.
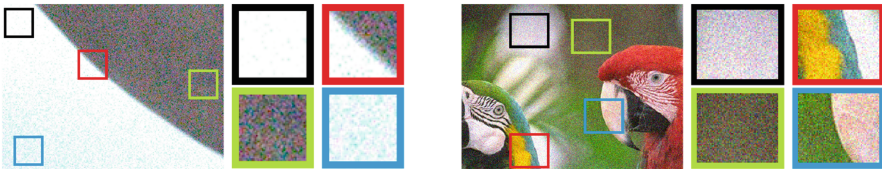

Fig. 1. Comparison of noisy image patches in real-world noisy image (left) and synthetic noisy image with additive white Gaussian noise (right). (Color figure online)

In the past decade, several denoising methods for real-world noisy images have been developed [21–26]. Liu et al. [21] proposed to estimate the noise via a "noise level function" and remove the noise for each channel of the real image. However, processing each channel separately would often achieve unsatisfactory performance and generate artifacts [5]. The methods [22,23] perform image denoising by concatenating the patches of RGB channels into a vector. However, the concatenation does not consider the different noise statistics among different channels. Besides, the method of [23] models complex noise via mixture
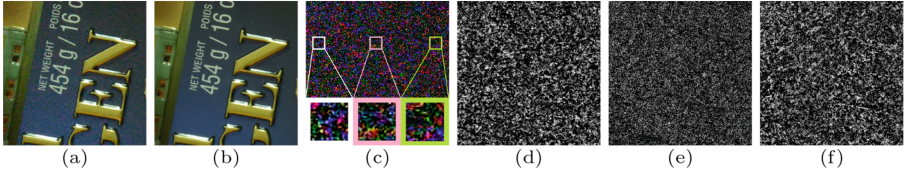
(a)   (b)   (c)   (d)   (e)   (f)

Fig. 2. An example of realistic noise. (a) A real-world noisy image captured by a Nikon D800 camera with ISO = 6400; (b) the "Ground Truth" image (please refer to Sect. 4.3) of (a); (c) difference between (a) and (b) (amplified for better illustration); (d)–(f) red, green, and blue channel of (c), respectively. The standard deviations (stds) of noise in the three boxes (white, pink, and green) plotted in (c) are 5.2, 6.5, and 3.3, respectively, while the stds of noise in each channel (d), (e), and (f) are 5.8, 4.4, and 5.5, respectively. (Color figure online)

of Gaussian distribution, which is time-consuming due to the use of variational Bayesian inference techniques. The method of [24] models the noise in a noisy image by a multivariate Gaussian and performs denoising by the Bayesian non-local means [30]. The commercial software Neat Image [25] estimates the global noise parameters from a flat region of the given noisy image and filters the noise accordingly. However, both the two methods [24,25] ignore the local statistical property of the noise which is signal dependent and varies with different pixels. The method [26] considers the different noise statistics in different channels, but ignores that the noise is signal dependent and has different levels in different local patches. By far, real-world image denoising is still a challenging problem in low level vision [29].

Sparse coding (SC) has been well studied in many computer vision and pattern recognition problems [31–33], including image denoising [4,5,10,11,14]. In general, given an input signal $\mathbf{y}$ and the dictionary $\mathbf{D}$ of coding atoms, the SC model can be formulated as

$$\min_{\mathbf{c}} \|\mathbf{y} - \mathbf{D}\mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_q, \tag{1}$$

where $\mathbf{c}$ is the coding vector of the signal $\mathbf{y}$ over the dictionary $\mathbf{D}$, λ is the regularization parameter, and $q = 0$ or 1 to enforce sparse regularization on $\mathbf{c}$. Some representative SC based image denoising methods include K-SVD [4], LSSC [10], and NCSR [11]. Though being effective on dealing with AWGN, SC based denoising methods are essentially limited by the data-fidelity term described by ℓ_2 (or Frobenius) norm, which actually assumes white Gaussian noise and is not able to characterize the signal dependent and realistic noise.

In this paper, we propose to lift the SC model (1) to a robust denoiser for real-world noisy images by utilizing the channel-wise statistics and locally signal dependent property of the realistic noise, as demonstrated in Fig. 2. Specifically, we propose a trilateral weighted sparse coding (TWSC) scheme for real-world image denoising. Two weight matrices are introduced into the data-fidelity term of the SC model to characterize the realistic noise property, and another weight matrix is introduced into the regularization term to characterize the sparsity

priors of natural images. We reformulate the proposed TWSC scheme into a linear equality-constrained optimization program, and solve it under the alternating direction method of multipliers (ADMM) [34] framework. One step of our ADMM is to solve a Sylvester equation, whose unique solution is not always guaranteed. Hence, we provide theoretical analysis on the existence and uniqueness of the solution to the proposed TWSC scheme. Experiments on three datasets of real-world noisy images demonstrate that the proposed TWSC scheme achieves much better performance than the state-of-the-art denoising methods.

2   The Proposed Real-World Image Denoising Algorithm

2.1   The Trilateral Weighted Sparse Coding Model

The real-world image denoising problem is to recover the clean image from its noisy observation. Current denoising methods [4–16] are mostly patch based. Given a noisy image, a local patch of size $p \times p \times 3$ is extracted from it and stretched to a vector, denoted by $\mathbf{y} = [\mathbf{y}_r^\top \ \mathbf{y}_g^\top \ \mathbf{y}_b^\top]^\top \in \mathbb{R}^{3p^2}$, where $\mathbf{y}_c \in \mathbb{R}^{p^2}$ is the corresponding patch in channel c, where $c \in \{r, g, b\}$ is the index of R, G, and B channels. For each local patch $\mathbf{y}$, we search the M most similar patches to it (including $\mathbf{y}$ itself) by Euclidean distance in a local window around it. By stacking the M similar patches column by column, we form a noisy patch matrix $\mathbf{Y} = \mathbf{X} + \mathbf{N} \in \mathbb{R}^{3p^2 \times M}$, where $\mathbf{X}$ and $\mathbf{N}$ are the corresponding clean and noise patch matrices, respectively. The noisy patch matrix can be written as $\mathbf{Y} = [\mathbf{Y}_r^\top \ \mathbf{Y}_g^\top \ \mathbf{Y}_b^\top]^\top$, where $\mathbf{Y}_c$ is the sub-matrix of channel c. Suppose that we have a dictionary $\mathbf{D} = [\mathbf{D}_r^\top \ \mathbf{D}_g^\top \ \mathbf{D}_b^\top]^\top$, where $\mathbf{D}_c$ is the sub-dictionary corresponding to channel c. In fact, the dictionary $\mathbf{D}$ can be learned from external natrual images, or from the input noisy patch matrix $\mathbf{Y}$.

Under the traditional sparse coding (SC) framework [35], the sparse coding matrix of $\mathbf{Y}$ over $\mathbf{D}$ can be obtained by

$$\hat{\mathbf{C}} = \arg \min_{\mathbf{C}} \|\mathbf{Y} - \mathbf{D}\mathbf{C}\|_F^2 + \lambda \|\mathbf{C}\|_1, \tag{2}$$

where λ is the regularization parameter. Once $\hat{\mathbf{C}}$ is computed, the latent clean patch matrix $\hat{\mathbf{X}}$ can be estimated as $\hat{\mathbf{X}} = \mathbf{D}\hat{\mathbf{C}}$. Though having achieved promising performance on additive white Gaussian noise (AWGN), the traditional SC based denoising methods [4,5,10,11,14] are very limited in dealing with realistic noise in real-world images captured by CCD or CMOS cameras. The reason is that the realistic noise is non-Gaussian, varies locally and across channels, which cannot be characterized well by the Frobenius norm in the SC model (2) [21,24,29,36].

To account for the varying statistics of realistic noise in different channels and different patches, we introduce two weight matrices $\mathbf{W}_1 \in \mathbb{R}^{3p^2 \times 3p^2}$ and $\mathbf{W}_2 \in \mathbb{R}^{M \times M}$ to characterize the SC residual $(\mathbf{Y} - \mathbf{D}\mathbf{C})$ in the data-fidelity term of Eq. (2). Besides, to better characterize the sparsity priors of the natural images, we introduce a third weight matrix $\mathbf{W}_3$, which is related to the distribution of

the sparse coefficients matrix $\mathbf{C}$, into the regularization term of Eq. (2). For the dictionary $\mathbf{D}$, we learn it adaptively by applying the SVD [37] to the given data matrix $\mathbf{Y}$ as

$$\mathbf{Y} = \mathbf{D}\mathbf{S}\mathbf{V}^\top. \tag{3}$$

Note that in this paper, we are not aiming at proposing a new dictionary learning scheme as [4] did. Once obtained from SVD, the dictionary $\mathbf{D}$ is fixed and not updated iteratively. Finally, the proposed trilateral weighted sparse coding (TWSC) model is formulated as:

$$\min_{\mathbf{C}} \|\mathbf{W}_1(\mathbf{Y} - \mathbf{D}\mathbf{C})\mathbf{W}_2\|_F^2 + \|\mathbf{W}_3^{-1}\mathbf{C}\|_1. \tag{4}$$

Note that the parameter λ has been implicitly incorporated into the weight matrix $\mathbf{W}_3$.

2.2   The Setting of Weight Matrices

In this paper, we set the three weight matrices $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{W}_3$ as diagonal matrices and grant clear physical meanings to them. $\mathbf{W}_1$ is a block diagonal matrix with three blocks, each of which has the same diagonal elements to describe the noise properties in the corresponding R, G, or B channel. Based on [29,36,38], the realistic noise in a local patch could be approximately modeled as Gaussian, and each diagonal element of $\mathbf{W}_2$ is used to describe the noise variance in the corresponding patch $\mathbf{y}$. Generally speaking, $\mathbf{W}_1$ is employed to regularize the row discrepancy of residual matrix $(\mathbf{Y} - \mathbf{D}\mathbf{C})$, while $\mathbf{W}_2$ is employed to regularize the column discrepancy of $(\mathbf{Y} - \mathbf{D}\mathbf{C})$. For matrix $\mathbf{W}_3$, each diagonal element is set based on the sparsity priors on $\mathbf{C}$.

We determine the three weight matrices $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{W}_3$ by employing the *Maximum A-Posterior* (MAP) estimation technique:

$$\hat{\mathbf{C}} = \arg\max_{\mathbf{C}} \ln P(\mathbf{C}|\mathbf{Y}) = \arg\max_{\mathbf{C}}\{\ln P(\mathbf{Y}|\mathbf{C}) + \ln P(\mathbf{C})\}. \tag{5}$$

The log-likelihood term $\ln P(\mathbf{Y}|\mathbf{C})$ is characterized by the statistics of noise. According to [29,36,38], it can be assumed that the noise is independently and identically distributed (i.i.d.) in each channel and each patch with Gaussian distribution. Denote by $\mathbf{y}_{cm}$ and $\mathbf{c}_m$ the mth column of the matrices $\mathbf{Y}_c$ and $\mathbf{C}$, respectively, and denote by σ_{cm} the noise std of $\mathbf{y}_{cm}$. We have

$$P(\mathbf{Y}|\mathbf{C}) = \prod_{c\in\{r,g,b\}} \prod_{m=1}^{M} (\pi\sigma_{cm})^{-p^2} e^{-\sigma_{cm}^{-2}\|\mathbf{y}_{cm}-\mathbf{D}_c\mathbf{c}_m\|_2^2}. \tag{6}$$

From the perspective of statistics [39], the set of $\{\sigma_{cm}\}$ can be viewed as a $3 \times M$ contingency table created by two variables σ_c and σ_m, and their relationship could be modeled by a log-linear model $\sigma_{cm} = \sigma_c^{l_1}\sigma_m^{l_2}$, where $l_1 + l_2 = 1$. Here we consider $\{\sigma_c\}, \{\sigma_m\}$ of equal importance and empirically set $l_1 = l_2 = 1/2$. The estimation of $\{\sigma_{cm}\}$ can be transferred to the estimation of $\{\sigma_c\}$ and $\{\sigma_m\}$, which will be introduced in the experimental section (Sect. 4).

The sparsity prior is imposed on the coefficients matrix $\mathbf{C}$, we assume that each column $\mathbf{c}_m$ of $\mathbf{C}$ follows i.i.d. Laplacian distribution. Specifically, for each entry $\mathbf{c}_m^i$, which is the coding coefficient of the mth patch $\mathbf{y}_m$ over the ith atom of dictionary $\mathbf{D}$, we assume that it follows distribution of $(2\mathbf{S}_i)^{-1}\exp(-\mathbf{S}_i^{-1}|\mathbf{c}_m^i|)$, where $\mathbf{S}_i$ is the ith diagonal element of the singular value matrix $\mathbf{S}$ in Eq. (3). Note that we set the scale factor of the distribution as the inverse of the ith singular value $\mathbf{S}_i$. This is because the larger the singular value $\mathbf{S}_i$ is, the more important the ith atom (i.e., singular vector) in $\mathbf{D}$ should be, and hence the distribution of the coding coefficients over this singular vector should have stronger regularization with weaker sparsity. The prior term in Eq. (5) becomes

$$P(\mathbf{C}) = \prod_{m=1}^{M} \prod_{i=1}^{3p^2} (2\mathbf{S}_i)^{-1} e^{-\mathbf{S}_i^{-1}|\mathbf{c}_m^i|}. \tag{7}$$

Put (7) and (6) into (5) and consider the log-linear model $\sigma_{cm} = \sigma_c^{1/2}\sigma_m^{1/2}$, we have

$$
\begin{aligned}
\hat{\mathbf{C}} &= \arg\min_{\mathbf{C}} \sum_{c\in\{r,g,b\}} \sum_{m=1}^{M} \sigma_{cm}^{-2} \|\mathbf{y}_{cm} - \mathbf{D}_c\mathbf{c}_m\|_2^2 + \sum_{m=1}^{M} \|\mathbf{S}^{-1}\mathbf{c}_m\|_1 \\
&= \arg\min_{\mathbf{C}} \sum_{c\in\{r,g,b\}} \sigma_c^{-1} \|(\mathbf{Y}_c - \mathbf{D}_c\mathbf{C})\mathbf{W}_2\|_F^2 + \|\mathbf{S}^{-1}\mathbf{C}\|_1 \\
&= \arg\min_{\mathbf{C}} \|\mathbf{W}_1(\mathbf{Y} - \mathbf{DC})\mathbf{W}_2\|_F^2 + \|\mathbf{W}_3^{-1}\mathbf{C}\|_1,
\end{aligned} \tag{8}
$$

where

$$
\begin{aligned}
\mathbf{W}_1 &= \mathrm{diag}(\sigma_r^{-1/2}\mathbf{I}_{p^2}, \sigma_g^{-1/2}\mathbf{I}_{p^2}, \sigma_b^{-1/2}\mathbf{I}_{p^2}), \\
\mathbf{W}_2 &= \mathrm{diag}(\sigma_1^{-1/2}, ..., \sigma_M^{-1/2}), \mathbf{W}_3 = \mathbf{S},
\end{aligned} \tag{9}
$$

and $\mathbf{I}_{p^2}$ is the p^2 dimensional identity matrix. Note that the diagonal elements of $\mathbf{W}_1$ and $\mathbf{W}_2$ are determined by the noise standard deviations in the corresponding channels and patches, respectively. The stronger the noise in a channel and a patch, the less that channel and patch will contribute to the denoised output.

2.3   Model Optimization

Letting $\mathbf{C}^* = \mathbf{W}_3^{-1}\mathbf{C}$, we can transfer the weight matrix $\mathbf{W}_3$ into the data-fidelity term of (4). Thus, the TWSC scheme (4) is reformulated as

$$\min_{\mathbf{C}^*} \|\mathbf{W}_1(\mathbf{Y} - \mathbf{DW}_3\mathbf{C}^*)\mathbf{W}_2\|_F^2 + \|\mathbf{C}^*\|_1. \tag{10}$$

To make the notation simple, we remove the superscript $*$ in $\mathbf{C}^*$ and still use $\mathbf{C}$ in the following development. We employ the variable splitting method [40] to solve the problem (10). By introducing an augmented variable $\mathbf{Z}$, the problem (10) is reformulated as a linear equality-constrained problem with two variables $\mathbf{C}$ and $\mathbf{Z}$:

$$\min_{\mathbf{C},\mathbf{Z}} \|\mathbf{W}_1(\mathbf{Y} - \mathbf{DW}_3\mathbf{C})\mathbf{W}_2\|_F^2 + \|\mathbf{Z}\|_1 \text{ s.t. } \mathbf{C} = \mathbf{Z}. \tag{11}$$

Since the objective function is separable w.r.t. the two variables, the problem (11) can be solved under the alternating direction method of multipliers (ADMM) [34] framework. The augmented Lagrangian function of (11) is:

$$\mathcal{L}(\mathbf{C}, \mathbf{Z}, \boldsymbol{\Delta}, \rho) = \|\mathbf{W}_1(\mathbf{Y} - \mathbf{D}\mathbf{W}_3\mathbf{C})\mathbf{W}_2\|_F^2 + \|\mathbf{Z}\|_1 + \langle \boldsymbol{\Delta}, \mathbf{C} - \mathbf{Z}\rangle + \frac{\rho}{2}\|\mathbf{C} - \mathbf{Z}\|_F^2,$$
(12)

where $\boldsymbol{\Delta}$ is the augmented Lagrangian multiplier and $\rho > 0$ is the penalty parameter. We initialize the matrix variables $\mathbf{C}_0$, $\mathbf{Z}_0$, and $\boldsymbol{\Delta}_0$ to be comfortable zero matrices and $\rho_0 > 0$. Denote by $(\mathbf{C}_k, \mathbf{Z}_k)$ and $\boldsymbol{\Delta}_k$ the optimization variables and Lagrange multiplier at iteration k ($k = 0, 1, 2, ...$), respectively. By taking derivatives of the Lagrangian function $\mathcal{L}$ w.r.t. $\mathbf{C}$ and $\mathbf{Z}$, and setting the derivatives to be zeros, we can alternatively update the variables as follows:

(1) **Update C by fixing Z and $\boldsymbol{\Delta}$:**

$$\mathbf{C}_{k+1} = \arg\min_{\mathbf{C}} \|\mathbf{W}_1(\mathbf{Y} - \mathbf{D}\mathbf{W}_3\mathbf{C})\mathbf{W}_2\|_F^2 + \frac{\rho_k}{2}\|\mathbf{C} - \mathbf{Z}_k + \rho_k^{-1}\boldsymbol{\Delta}_k\|_F^2.$$
(13)

This is a two-sided weighted least squares regression problem with the solution satisfying that

$$\mathbf{A}\mathbf{C}_{k+1} + \mathbf{C}_{k+1}\mathbf{B}_k = \mathbf{E}_k,$$
(14)

where

$$\mathbf{A} = \mathbf{W}_3^\top \mathbf{D}^\top \mathbf{W}_1^\top \mathbf{W}_1 \mathbf{D} \mathbf{W}_3, \mathbf{B}_k = \frac{\rho_k}{2}(\mathbf{W}_2\mathbf{W}_2^\top)^{-1},$$
$$\mathbf{E}_k = \mathbf{W}_3^\top \mathbf{D}^\top \mathbf{W}_1^\top \mathbf{W}_1 \mathbf{Y} + (\frac{\rho_k}{2}\mathbf{Z}_k - \frac{1}{2}\boldsymbol{\Delta}_k)(\mathbf{W}_2\mathbf{W}_2^\top)^{-1}.$$
(15)

Equation (14) is a standard Sylvester equation (SE) which has a unique solution if and only if $\sigma(\mathbf{A}) \cap \sigma(-\mathbf{B}_k) = \emptyset$, where $\sigma(\mathbf{F})$ denotes the spectrum, i.e., the set of eigenvalues, of the matrix $\mathbf{F}$ [41]. We can rewrite the SE (14) as

$$(\mathbf{I}_M \otimes \boldsymbol{A} + \mathbf{B}_k^\top \otimes \boldsymbol{I}_{3p^2})\text{vec}(\mathbf{C}_{k+1}) = \text{vec}(\mathbf{E}_k),$$
(16)

and the solution $\mathbf{C}_{k+1}$ (if existed) can be obtained via $\mathbf{C}_{k+1} = \text{vec}^{-1}(\text{vec}(\mathbf{C}_{k+1}))$, where $\text{vec}^{-1}(\bullet)$ is the inverse of the vec-operator $\text{vec}(\bullet)$. Detailed theoretical analysis on the existence of the unique solution is given in Sect. 3.1.

(2) **Update Z by fixing C and $\boldsymbol{\Delta}$:**

$$\mathbf{Z}_{k+1} = \arg\min_{\mathbf{Z}} \frac{\rho_k}{2}\|\mathbf{Z} - (\mathbf{C}_{k+1} + \rho_k^{-1}\boldsymbol{\Delta}_k)\|_F^2 + \|\mathbf{Z}\|_1.$$
(17)

This problem has a closed-form solution as

$$\mathbf{Z}_{k+1} = \mathcal{S}_{\rho_k^{-1}}(\mathbf{C}_{k+1} + \rho_k^{-1}\boldsymbol{\Delta}_k),$$
(18)

where $\mathcal{S}_\lambda(x) = \text{sign}(x) \cdot \max(x - \lambda, 0)$ is the soft-thresholding operator.

Algorithm 1: Solve the TWSC Model (4) via ADMM

Input: Y, W$_1$, W$_2$, W$_3$, μ, Tol, K_1;

Initialization: C$_0$ = Z$_0$ = $\boldsymbol{\Delta}_0$ = 0, $\rho_0 > 0$, $k = 0$, T = False;

While (T == false) **do**

1. Update **C**$_{k+1}$ by solving Eq. (13);
2. Update **Z**$_{k+1}$ by soft thresholding (18);
3. Update $\boldsymbol{\Delta}_{k+1}$ by Eq. (19);
4. Update ρ_{k+1} by $\rho_{k+1} = \mu\rho_k$, where $\mu \geq 1$;
5. $k \leftarrow k + 1$;
      if (Converged) or ($k \geq K_1$)
6.   T $\leftarrow$ True;
      end if
end while

Output: Matrices **C** and **Z**.

---

(3) **Update $\boldsymbol{\Delta}$ by fixing X and Z:**

$$\boldsymbol{\Delta}_{k+1} = \boldsymbol{\Delta}_k + \rho_k(\mathbf{C}_{k+1} - \mathbf{Z}_{k+1}). \tag{19}$$

(4) **Update ρ**: $\rho_{k+1} = \mu\rho_k$, where $\mu \geq 1$.

The above alternative updating steps are repeated until the convergence condition is satisfied or the number of iterations exceeds a preset threshold K_1. The ADMM algorithm converges when $\|\mathbf{C}_{k+1} - \mathbf{Z}_{k+1}\|_F \leq$ Tol, $\|\mathbf{C}_{k+1} - \mathbf{C}_k\|_F \leq$ Tol, and $\|\mathbf{Z}_{k+1} - \mathbf{Z}_k\|_F \leq$ Tol are simultaneously satisfied, where Tol > 0 is a small tolerance number. We summarize the updating procedures in Algorithm 1.

Convergence Analysis. The convergence of Algorithm 1 can be guaranteed since the overall objective function (11) is convex with a global optimal solution. In Fig. 3, we can see that the maximal values in $|\mathbf{C}_{k+1} - \mathbf{Z}_{k+1}|$, $|\mathbf{C}_{k+1} - \mathbf{C}_k|$, $|\mathbf{Z}_{k+1} - \mathbf{Z}_k|$ approach to 0 simultaneously in 50 iterations.
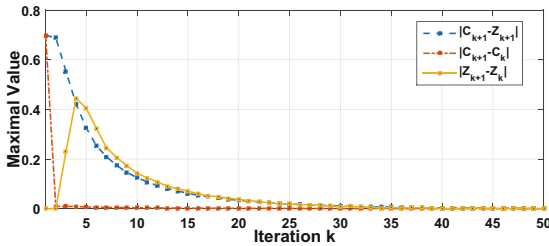

Fig. 3. The convergence curves of maximal values in entries of $|\mathbf{C}_{k+1} - \mathbf{Z}_{k+1}|$ (blue line), $|\mathbf{C}_{k+1} - \mathbf{C}_k|$ (red line), and $|\mathbf{Z}_{k+1} - \mathbf{Z}_k|$ (yellow line). The test image is the image in Fig. 2(a). (Color figure online)

2.4 The Denoising Algorithm

Given a noisy color image, suppose that we have extracted N local patches $\{\mathbf{y}_j\}_{j=1}^N$ and their similar patches. Then N noisy patch matrices $\{\mathbf{Y}_j\}_{j=1}^N$ can

be formed to estimate the clean patch matrices $\{\mathbf{X}_j\}_{j=1}^N$. The patches in matrices $\{\mathbf{X}_j\}_{j=1}^N$ are aggregated to form the denoised image $\hat{\mathbf{x}}_c$. To obtain better denoising results, we perform the above denoising procedures for several (e.g., K_2) iterations. The proposed TWSC scheme based real-world image denoising algorithm is summarized in Algorithm 2.

Algorithm 2: Image Denoising by TWSC

Input: Noisy image $\mathbf{y}_c$, $\{\sigma_r, \sigma_g, \sigma_b\}$, K_2;
Initialization: $\hat{\mathbf{x}}_c^{(0)} = \mathbf{y}_c$, $\mathbf{y}_c^{(0)} = \mathbf{y}_c$;
for $k = 1 : K_2$ **do**
1. Set $\mathbf{y}_c^{(k)} = \hat{\mathbf{x}}_c^{(k-1)}$;
2. Extract local patches $\{\mathbf{y}_j\}_{j=1}^N$ from $\mathbf{y}_c^{(k)}$;
   for each patch $\mathbf{y}_j$ **do**
3.   Search nonlocal similar patches $\mathbf{Y}_j$;
4.   Apply the TWSC scheme (4) to $\mathbf{Y}_j$ and obtain the estimated $\mathbf{X}_j = \mathbf{DC}$;
   end for
5. Aggregate $\{\mathbf{X}_j\}_{j=1}^N$ to form the image $\hat{\mathbf{x}}_c^{(k)}$;
end for
Output: Denoised image $\hat{\mathbf{x}}_c^{(K_2)}$.

---

3 Existence and Faster Solution of Sylvester Equation

The solution of the Sylvester equation (SE) (14) does not always exist, though the solution is unique if it exists. Besides, solving SE (14) is usually computationally expensive in high dimensional cases. In this section, we provide a sufficient condition to guarantee the existence of the solution to SE (14), as well as a faster solution of (14) to save the computational cost of Algorithms 1 and 2.

3.1 Existence of the Unique Solution

Before we prove the existence of unique solution of SE (14), we first introduce the following theorem.

Theorem 1. *Assume that $\mathbf{A} \in \mathbb{R}^{3p^2 \times 3p^2}$, $\mathbf{B} \in \mathbb{R}^{M \times M}$ are both symmetric and positive semi-definite matrices. If at least one of $\mathbf{A}, \mathbf{B}$ is positive definite, the Sylvester equation $\mathbf{AC} + \mathbf{CB} = \mathbf{E}$ has a unique solution for $\mathbf{C} \in \mathbb{R}^{3p^2 \times M}$.*

The proof of Theorem 1 can be found in the supplementary file. Then we have the following corollary.

Corollary 1. *The SE (14) has a unique solution.*

Proof. Since $\mathbf{A}, \mathbf{B}_k$ in (14) are both symmetric and positive definite matrices, according to Theorem 1, the SE (14) has a unique solution.

3.2   Faster Solution

The solution of the SE (14) is typically obtained by the Bartels-Stewart algorithm [42]. This algorithm firstly employs a QR factorization [43], implemented via Gram-Schmidt process, to decompose the matrices $\mathbf{A}$ and $\mathbf{B}_k$ into Schur forms, and then solves the obtained triangular system by the back-substitution method [44]. However, since the matrices $\mathbf{I}_M \otimes \boldsymbol{A}$ and $\mathbf{B}_k^\top \otimes \boldsymbol{I}_{3p^2}$ are of $3p^2M \times 3p^2M$ dimensions, it is computationally expensive ($\mathcal{O}(p^6M^3)$) to calculate their QR factorization to obtain the Schur forms. By exploiting the specific properties of our problem, we provide a faster while exact solution for the SE (14).

Since the matrices $\mathbf{A}, \mathbf{B}_k$ in (14) are symmetric and positive definite, the matrix $\mathbf{A}$ can be eigen-decomposed as $\mathbf{A} = \mathbf{U_A} \boldsymbol{\Sigma_A} \mathbf{U_A^\top}$, with computational cost of $\mathcal{O}(p^6)$. Left multiply both sides of the SE (14) by $\mathbf{U_A^\top}$, we can get $\boldsymbol{\Sigma}_A \mathbf{U_A^\top} \mathbf{C}_{k+1} + \mathbf{U_A^\top} \mathbf{C}_{k+1} \mathbf{B}_k = \mathbf{U_A^\top} \mathbf{E}_k$. This can be viewed as an SE w.r.t. the matrix $\mathbf{U_A^\top} \mathbf{C}_{k+1}$, with a unique solution $\text{vec}(\mathbf{U_A^\top} \mathbf{C}_{k+1}) = (\mathbf{I}_M \otimes \boldsymbol{\Sigma_A} + \mathbf{B}_k^\top \otimes \boldsymbol{I}_{3p^2})^{-1} \text{vec}(\mathbf{U_A^\top} \mathbf{E}_k)$. Since the matrix $(\mathbf{I}_M \otimes \boldsymbol{\Sigma_A} + \mathbf{B}_k^\top \otimes \boldsymbol{I}_{3p^2})$ is diagonal and positive definite, its inverse can be calculated on each diagonal element of $(\mathbf{I}_M \otimes \boldsymbol{\Sigma_A} + \mathbf{B}_k^\top \otimes \boldsymbol{I}_{3p^2})$. The computational cost for this step is $\mathcal{O}(p^2M)$. Finally, the solution $\mathbf{C}_{k+1}$ can be obtained via $\mathbf{C}_{k+1} = \mathbf{U_A} \text{vec}^{-1}(\text{vec}(\mathbf{U_A^\top} \mathbf{C}_{k+1}))$. By this way, the complexity for solving the SE (14) is reduced from $\mathcal{O}(p^6M^3)$ to $\mathcal{O}(\max(p^6, p^2M))$, which is a huge computational saving.

4   Experiments

To validate the effectiveness of our proposed TWSC scheme, we apply it to both synthetic additive white Gaussian noise (AWGN) corrupted images and real-world noisy images captured by CCD or CMOS cameras. To better demonstrate the roles of the three weight matrices in our model, we compare with a baseline method, in which the weight matrices $\mathbf{W}_1, \mathbf{W}_2$ are set as comfortable identity matrices, while the matrix $\mathbf{W}_3$ is set as in (8). We call this baseline method the *Weighted Sparse Coding* (WSC).

4.1   Experimental Settings

Noise Level Estimation. For most image denoising algorithms, the standard deviation (std) of noise should be given as a parameter. In this work, we provide an exploratory approach to solve this problem. Specifically, the noise std σ_c of channel c can be estimated by some noise estimation methods [45–47]. In Algorithm 2, the noise std for the mth patch of $\mathbf{Y}$ can be initialized as

$$\sigma_m = \sigma \triangleq \sqrt{(\sigma_r^2 + \sigma_g^2 + \sigma_b^2)/3} \tag{20}$$

and updated in the following iterations as

$$\sigma_m = \sqrt{\max(0, \sigma^2 - \|\mathbf{y}_m - \mathbf{x}_m\|_2^2)}, \tag{21}$$

where $\mathbf{y}_m$ is the mth column in the patch matrix $\mathbf{Y}$, and $\mathbf{x}_m = \mathbf{D}\mathbf{c}_m$ is the mth patch recovered in previous iteration (please refer to Sect. 2.4).

Implementation Details. We empirically set the parameter $\rho_0 = 0.5$ and $\mu = 1.1$. The maximum number of iteration is set as $K_1 = 10$. The window size for similar patch searching is set as 60×60. For parameters p, M, K_2, we set $p = 7$, $M = 70$, $K_2 = 8$ for $0 < \sigma \le 20$; $p = 8$, $M = 90$, $K_2 = 12$ for $20 < \sigma \le 40$; $p = 8$, $M = 120$, $K_2 = 12$ for $40 < \sigma \le 60$; $p = 9$, $M = 140$, $K_2 = 14$ for $60 < \sigma \le 100$. All parameters are fixed in our experiments. We will release the code with the publication of this work.

Table 1. Average results of PSNR(dB) and SSIM of different denoising algorithms on 20 grayscale images corrupted by AWGN noise.

σ_n	Metric	BM3D-SAPCA	LSSC	NCSR	WNNM	TNRD	DnCNN	WSC	TWSC
15	PSNR	32.42	32.27	32.19	32.43	32.27	32.59	32.06	32.34
	SSIM	0.8860	0.8849	0.8814	0.8841	0.8815	0.8879	0.8673	0.8846
25	PSNR	30.02	29.84	29.76	30.05	29.87	30.22	29.57	29.98
	SSIM	0.8364	0.8329	0.8293	0.8365	0.8314	0.8415	0.8179	0.8372
35	PSNR	28.48	28.26	28.17	28.51	28.33	28.66	28.01	28.49
	SSIM	0.7969	0.7908	0.7855	0.7958	0.7907	0.8021	0.7765	0.7987
50	PSNR	26.85	26.64	26.55	26.92	26.75	27.08	26.35	26.93
	SSIM	0.7481	0.7405	0.7391	0.7499	0.7415	0.7563	0.7258	0.7530
75	PSNR	24.74	24.77	24.66	25.15	24.97	25.24	24.54	25.15
	SSIM	0.6649	0.6746	0.6793	0.6903	0.6801	0.6931	0.6612	0.6949

4.2 Results on AWGN Noise Removal

We first compare the proposed TWSC scheme with the leading AWGN denoising methods such as BM3D-SAPCA [9] (which usually performs better than BM3D [7]), LSSC [10], NCSR [11], WNNM [13], TNRD [19], and DnCNN [20] on 20 grayscale images commonly used in [7]. Note that TNRD and DnCNN are both discriminative learning based methods, and we use the models trained originally by the authors. Each noisy image is generated by adding the AWGN noise to the clean image, while the std of the noise is set as $\sigma \in \{15, 25, 35, 50, 75\}$ in this paper. Note that in this experiment we set the weight matrix $\mathbf{W}_1 = \sigma^{-1/2}\mathbf{I}_{p^2}$ since the input images are grayscale.

The averaged PSNR and SSIM [48] results are listed in Table 1. One can see that the proposed TWSC achieves comparable performance with WNNM, TNRD and DnCNN in most cases. It should be noted that TNRD and DnCNN are trained on clean and synthetic noisy image pairs, while TWSC only utilizes the tested noisy image. Besides, one can see that the proposed TWSC works much better than the baseline method WSC, which proves that the weight matrix $\mathbf{W}_2$ can characterize better the noise statistics in local image patches. Due to limited space, we leave the visual comparisons of different methods in the supplementary file.

4.3   Results on Realistic Noise Removal

We evaluate the proposed TWSC scheme on three publicly available real-world noisy image datasets [24, 29, 49].

   Dataset 1 is provided in [49], which includes around 20 real-world noisy images collected under uncontrolled environment. Since there is no "ground truth" of the noisy images, we only compare the visual quality of the denoised images by different methods.

   Dataset 2 is provided in [24], which includes noisy images of 11 static scenes captured by Canon 5D Mark 3, Nikon D600, and Nikon D800 cameras. The real-world noisy images were collected under controlled indoor environment. Each scene was shot 500 times under the same camera and camera setting. The mean image of the 500 shots is roughly taken as the "ground truth", with which the PSNR and SSIM [48] can be computed. 15 images of size 512×512 were cropped to evaluate different denoising methods. Recently, some other datasets such as [50] are also constructed by employing the strategies of this dataset.

   Dataset 3 is called the Darmstadt Noise Dataset (DND) [29], which includes 50 different pairs of images of the same scenes captured by Sony A7R, Olympus E-M10, Sony RX100 IV, and Huawei Nexus 6P. The real-world noisy images are collected under higher ISO values with shorter exposure time, while the "ground truth" images are captured under lower ISO values with adjusted longer exposure times. Since the captured images are of megapixel-size, the authors cropped 20 bounding boxes of 512×512 pixels from each image in the dataset, yielding 1000 test crops in total. However, the "ground truth" images are not open access, and we can only submit the denoising results to the authors' Project Website and get the PSNR and SSIM [48] results.

Comparison Methods. We compare the proposed TWSC method with CBM3D [8], TNRD [19], DnCNN [20], the commercial software Neat Image (NI) [25], the state-of-the-art real image denoising methods "Noise Clinic" (NC) [22], CC [24], and MCWNNM [26]. We also compare with the baseline method WSC described in Sect. 4 as a baseline. The methods of CBM3D and DnCNN can directly deal with color images, and the input noise std is set by Eq. (20). For TNRD, MCWNNM, and TWSC, we use [46] to estimate the noise std σ_c ($c \in \{r, g, b\}$) for each channel. For blind mode DnCNN, we use its color version provided by the authors and there is no need to estimate the noise std. Since TNRD is designed for grayscale images, we applied them to each channel of real-world noisy images. TNRD achieves its best results when setting the noise std of the trained models at $\sigma_c = 10$ on these datasets.

Results on Dataset 1. Figure 4 shows the denoised images of "Dog" (the method CC [24] is not compared since its testing code is not available). One can see that CBM3D, TNRD, DnCNN, NI and NC generate some noise-caused color artifacts across the whole image, while MCWNNM and WSC tend to over-smooth a little the image. The proposed TWSC removes more clearly the noise without over-smoothing much the image details. These results demonstrate that the methods designed for AWGN are not effective for realistic noise
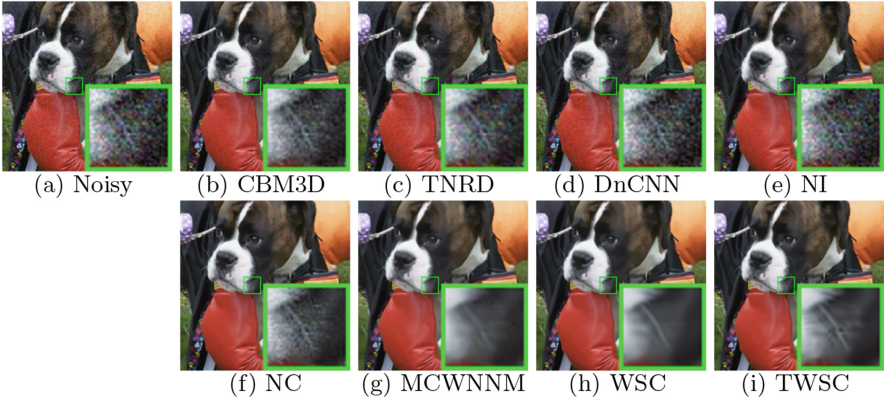
(a) Noisy   (b) CBM3D   (c) TNRD   (d) DnCNN   (e) NI

(f) NC   (g) MCWNNM   (h) WSC   (i) TWSC

Fig. 4. Denoised images of the real noisy image *Dog* [49] by different methods. Note that the ground-truth clean image of the noisy input is not available.

removal. Though NC and NI methods are specifically developed for real-world noisy images, their performance is not satisfactory. In comparison, the proposed TWSC works much better in removing the noise while maintaining the details (see the zoom-in window in "Dog") than the other competing methods. More visual comparisons can be found in the supplementary file.

Results on Dataset 2. The average PSNR and SSIM results on the 15 cropped images by competing methods are listed in Table 2. One can see that the proposed TWSC is much better than other competing methods, including the baseline method WSC and the recently proposed CC, MCWNNM. Figure 5 shows the denoised images of a scene captured by Nikon D800 at ISO = 6400. One can see that the proposed TWSC method results in not only higher PSNR and SSIM measures, but also much better visual quality than other methods. Due to limited space, we do not show the results of baseline method WSC in visual quality comparison. More visual comparisons can be found in the supplementary file.
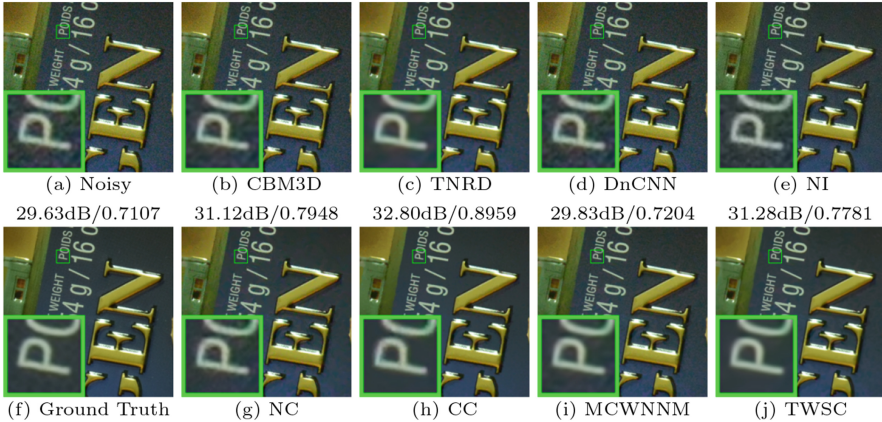
Table 2. Average results of PSNR(dB) and SSIM of different denoising methods on 15 cropped real-world noisy images used in [24].

	CBM3D	TNRD	DnCNN	NI	NC	CC	MCWNNM	WSC	TWSC
PSNR	35.19	36.61	33.86	35.49	36.43	36.88	37.70	37.36	**37.81**
SSIM	0.8580	0.9463	0.8635	0.9126	0.9364	0.9481	0.9542	0.9516	**0.9586**

Results on Dataset 3. In Table 3, we list the average PSNR and SSIM results of the competing methods on the 1000 cropped images in the DND dataset [29]. We can see again that the proposed TWSC achieves much better performance than the other competing methods. Note that the "ground truth" images of this dataset have not been published, but one can submit the denoised images to the project website and get the PSNR and SSIM results. More results can

Table 3. Average results of PSNR(dB) and SSIM of different denoising methods on 1000 cropped real-world noisy images in [29].

	CBM3D	TNRD	DnCNN	NI	NC	MCWNNM	WSC	TWSC
PSNR	32.14	34.15	32.41	35.11	36.07	37.38	36.81	**37.94**
SSIM	0.7773	0.8271	0.7897	0.8778	0.9013	0.9294	0.9165	**0.9403**

(a) Noisy	(b) CBM3D	(c) TNRD	(d) DnCNN	(e) NI
29.63dB/0.7107	31.12dB/0.7948	32.80dB/0.8959	29.83dB/0.7204	31.28dB/0.7781

| (f) Ground Truth | (g) NC | (h) CC | (i) MCWNNM | (j) TWSC |

Fig. 5. Denoised images of the real noisy image *Nikon D800 ISO 6400 1* [24] by different methods. This scene was shot 500 times under the same camera and camera setting. The mean image of the 500 shots is roughly taken as the "Ground Truth".

be found in the website of the DND dataset (https://noise.visinf.tu-darmstadt.de/benchmark/#results_srgb). Figure 6 shows the denoised images of a scene captured by a Nexus 6P camera. One can still see that the proposed TWSC method results better visual quality than the other denoising methods. More visual comparisons can be found in the supplementary file.

Comparison on Speed. We compare the average computational time (second) of different methods (except CC) to process one 512×512 image on the DND Dataset [29]. The results are shown in Table 4. All experiments are run under the Matlab2014b environment on a machine with Intel(R) Core(TM) i7-5930K CPU of 3.5 GHz and 32 GB RAM. The fastest speed is highlighted in bold. One can see that Neat Image (NI) is the fastest and it spends about 1.1 s to process an image, while the proposed TWSC needs about 195 s. Noted that Neat Image is a highly-optimized software with parallelization, CBM3D, TNRD, and NC are implemented with compiled C++ mex-function and with parallelization, while DnCNN, MCWNNM, and the proposed WSC and TWSC are implemented purely in Matlab.
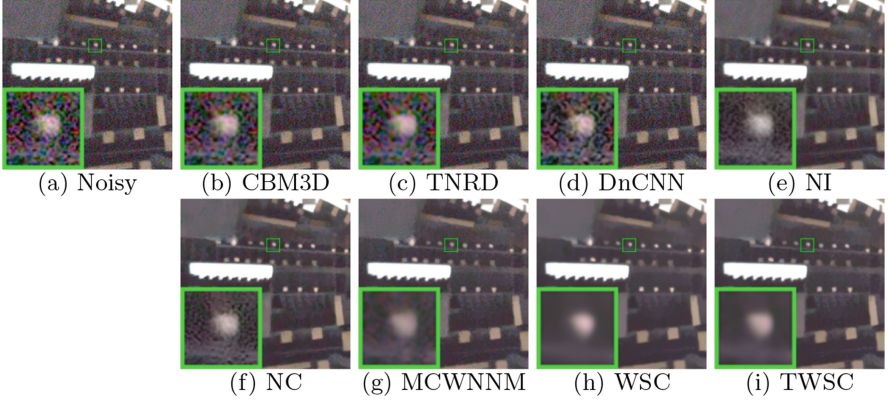
(a) Noisy   (b) CBM3D   (c) TNRD   (d) DnCNN   (e) NI

(f) NC   (g) MCWNNM   (h) WSC   (i) TWSC

Fig. 6. Denoised images of the real noisy image "*0001_2*" captured by Nexus 6P [29] by different methods. Note that the ground-truth clean image of the noisy input is not publicly released yet.

Table 4. Average computational time (s) of different methods to process a 512×512 image in the DND dataset [29].

	CBM3D	TNRD	DnCNN	NI	NC	MCWNNM	WSC	TWSC
Time	6.9	5.2	79.5	**1.1**	15.6	208.1	188.6	195.2

4.4 Visualization of the Weight Matrices

The three diagonal weight matrices in the proposed TWSC model (4) have clear physical meanings, and it is interesting to analyze how the matrices actually relate to the input image by visualizing the resulting matrices. To this end, we applied TWSC to the real-world (estimated noise stds of R/G/B: 11.4/14.8/18.4) and synthetic AWGN (std of all channels: 25) noisy images shown in Fig. 1. The final diagonal weight matrices for two typical patch matrices ($\boldsymbol{Y}$) from the two images are visualized in Fig. 7. One can see that the matrix $\boldsymbol{W}_1$ reflects well the noise levels in the images. Though matrix $\boldsymbol{W}_2$ is initialized as an identity matrix, it is changed in iterations since noise in different patches are removed differently. For real-world noisy images, the noise levels of different patches in $\boldsymbol{Y}$ are different, hence the elements of $\boldsymbol{W}_2$ vary a lot. In contrast, the noise levels of patches in the synthetic noisy image are similar, thus the elements of $\boldsymbol{W}_2$ are similar. The weight matrix $\boldsymbol{W}_3$ is basically determined by the patch structure but not noise, and we do not plot it here.
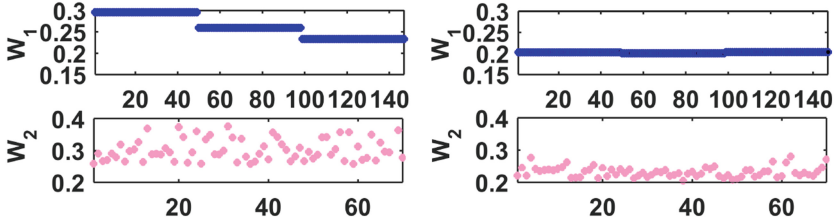
Fig. 7. Visualization of weight matrices W_1 and W_2 on the real-world noisy image (left) and the synthetic noisy image (right) shown in Fig. 1.

5 Conclusion

The realistic noise in real-world noisy images captured by CCD or CMOS cameras is very complex due to the various factors in digital camera pipelines, making the real-world image denoising problem much more challenging than additive white Gaussian noise removal. We proposed a novel trilateral weighted sparse coding (TWSC) scheme to exploit the noise properties across different channels and local patches. Specifically, we introduced two weight matrices into the datafidelity term of the traditional sparse coding model to adaptively characterize the noise statistics in each patch of each channel, and another weight matrix into the regularization term to better exploit sparsity priors of natural images. The proposed TWSC scheme was solved under the ADMM framework and the solution to the Sylvester equation is guaranteed. Experiments demonstrated the superior performance of TWSC over existing state-of-the-art denoising methods, including those methods designed for realistic noise in real-world noisy images.

References

1. Zhu, L., Fu, C.W., Brown, M.S., Heng, P.A.: A non-local low-rank framework for ultrasound speckle reduction. In: CVPR, pp. 5650–5658 (2017)
2. Granados, M., Kim, K., Tompkin, J., Theobalt, C.: Automatic noise modeling for ghost-free hdr reconstruction. ACM Trans. Graph. **32**(6), 1–10 (2013)
3. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: high confidence predictions for unrecognizable images. In: CVPR, pp. 427–436 (2015)
4. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans. Image Process. **15**(12), 3736–3745 (2006)
5. Mairal, J., Elad, M., Sapiro, G.: Sparse representation for color image restoration. IEEE Trans. Image Process. **17**(1), 53–69 (2008)
6. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: CVPR, pp. 60–65 (2005)
7. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. Image Process. **16**(8), 2080–2095 (2007)
8. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space. In: ICIP, pp. 313–316. IEEE (2007)

9. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: BM3D image denoising with shape-adaptive principal component analysis. In: SPARS (2009)
10. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Non-local sparse models for image restoration. In: ICCV, pp. 2272–2279 (2009)
11. Dong, W., Zhang, L., Shi, G., Li, X.: Nonlocally centralized sparse representation for image restoration. IEEE Trans. Image Process. **22**(4), 1620–1630 (2013)
12. Dong, W., Shi, G., Li, X.: Nonlocal image restoration with bilateral variance estimation: a low-rank approach. IEEE Trans. Image Process. **22**(2), 700–711 (2013)
13. Gu, S., Zhang, L., Zuo, W., Feng, X.: Weighted nuclear norm minimization with application to image denoising. In: CVPR, pp. 2862–2869. IEEE (2014)
14. Xu, J., Zhang, L., Zuo, W., Zhang, D., Feng, X.: Patch group based nonlocal self-similarity prior learning for image denoising. In ICCV, pp. 244–252 (2015)
15. Roth, S., Black, M.J.: Fields of experts. Int. J. Comput. Vis. **82**(2), 205–229 (2009)
16. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: ICCV, pp. 479–486 (2011)
17. Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: can plain neural networks compete with BM3D? In: CVPR, pp. 2392–2399 (2012)
18. Schmidt, U., Roth, S.: Shrinkage fields for effective image restoration. In: CVPR, pp. 2774–2781, June 2014
19. Chen, Y., Yu, W., Pock, T.: On learning optimized reaction diffusion processes for effective image restoration. In: CVPR, pp. 5261–5269 (2015)
20. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Trans. Image Process. **26**, 3142–3155 (2017)
21. Liu, C., Szeliski, R., Kang, S.B., Zitnick, C.L., Freeman, W.T.: Automatic estimation and removal of noise from a single image. IEEE TPAMI **30**(2), 299–314 (2008)
22. Lebrun, M., Colom, M., Morel, J.M.: Multiscale image blind denoising. IEEE Trans. Image Process. **24**(10), 3149–3161 (2015)
23. Zhu, F., Chen, G., Heng, P.A.: From noise modeling to blind image denoising. In: CVPR, June 2016
24. Nam, S., Hwang, Y., Matsushita, Y., Kim, S.J.: A holistic approach to cross-channel image noise modeling and its application to image denoising. In: CVPR, pp. 1683–1691 (2016)
25. ABSoft, N.: Neat Image. https://ni.neatvideo.com/home
26. Xu, J., Zhang, L., Zhang, D., Feng, X.: Multi-channel weighted nuclear norm minimization for real color image denoising. In: ICCV (2017)
27. Xu, J., Ren, D., Zhang, L., Zhang, D.: Patch group based Bayesian learning for blind image denoising. In: Chen, C.-S., Lu, J., Ma, K.-K. (eds.) ACCV 2016. LNCS, vol. 10116, pp. 79–95. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54407-6_6
28. Xu, J., Zhang, L., Zhang, D.: External prior guided internal prior learning for real-world noisy image denoising. IEEE Trans. Image Process. **27**(6), 2996–3010 (2018)
29. Plötz, T., Roth, S.: Benchmarking denoising algorithms with real photographs. In: CVPR (2017)
30. Kervrann, C., Boulanger, J., Coupé, P.: Bayesian non-local means filter, image redundancy and adaptive dictionaries for noise removal. In: Sgallari, F., Murli, A., Paragios, N. (eds.) SSVM 2007. LNCS, vol. 4485, pp. 520–532. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72823-8_45

31. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. IEEE TPAMI **31**(2), 210–227 (2009)
32. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR, pp. 1794–1801 (2009)
33. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution via sparse representation. IEEE Trans. Image Process. **19**(11), 2861–2873 (2010)
34. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**(1), 1–122 (2011)
35. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Society. Ser. B (Methodol.) **58**, 267–288 (1996)
36. Khashabi, D., Nowozin, S., Jancsary, J., Fitzgibbon, A.W.: Joint demosaicing and denoising via learned nonparametric random fields. IEEE Trans. Image Process. **23**(12), 4968–4981 (2014)
37. Eckart, C., Young, G.: The approximation of one matrix by another of lower rank. Psychometrika **1**(3), 211–218 (1936)
38. Leung, B., Jeon, G., Dubois, E.: Least-squares luma-chroma demultiplexing algorithm for bayer demosaicking. IEEE Trans. Image Process. **20**(7), 1885–1894 (2011)
39. McCullagh, P.: Generalized linear models. Eur. J. Oper. Res. **16**(3), 285–292 (1984)
40. Eckstein, J., Bertsekas, D.P.: On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Program. **55**(1), 293–318 (1992)
41. Simoncini, V.: Computational methods for linear matrix equations. SIAM Rev. **58**(3), 377–441 (2016)
42. Bartels, R.H., Stewart, G.W.: Solution of the matrix equation AX + XB = C. Commun. ACM **15**(9), 820–826 (1972)
43. Golub, G., Van Loan, C.: Matrix Computations, 3rd edn. Johns Hopkins University Press, Baltimore (1996)
44. Bareiss, E.: Sylvesters identity and multistep integer-preserving Gaussian elimination. Math. Comput. **22**(103), 565–578 (1968)
45. Liu, X., Tanaka, M., Okutomi, M.: Single-image noise level estimation for blind denoising. IEEE Trans. Image Process. **22**(12), 5226–5237 (2013)
46. Chen, G., Zhu, F., Pheng, A.H.: An efficient statistical method for image noise level estimation. In: ICCV, December 2015
47. Sutour, C., Deledalle, C.A., Aujol, J.F.: Estimation of the noise level function based on a nonparametric detection of homogeneous image regions. SIAM J. Imaging Sci. **8**(4), 2622–2661 (2015)
48. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
49. Lebrun, M., Colom, M., Morel, J.M.: The noise clinic: a blind image denoising algorithm. http://www.ipol.im/pub/art/2015/125/. Accessed 28 Jan 2015
50. Xu, J., Li, H., Liang, Z., Zhang, D., Zhang, L.: Real-world noisy image denoising: a new benchmark. CoRR abs/1804.02603 (2018)