Image denoising based on improved data-driven sparse representation

Dai-Qiang Chen

Abstract

Sparse representation of images under certain transform domain has been playing a fundamental role in image restoration tasks. One such representative method is the widely used wavelet tight frame systems. Instead of adopting fixed filters for constructing a tight frame to sparsely model any input image, a data-driven tight frame was proposed for the sparse representation of images, and shown to be very efficient for image denoising very recently. However, in this method the number of framelet filters used for constructing a tight frame is the same as the length of filters. In fact, through further investigation it is found that part of these filters are unnecessary and even harmful to the recovery effect due to the influence of noise. Therefore, an improved data-driven sparse representation systems constructed with much less number of filters are proposed. Numerical results on denoising experiments demonstrate that the proposed algorithm overall outperforms the original data-driven tight frame construction scheme on both the recovery quality and computational time.

Index Terms

data-driven; image denoising; sparse representation; signal subspace; noise subspace.

I. INTRODUCTION

Image denoising, which aims to estimate the clean image from its noise-corrupted observation, is a fundamental topic in the field of computer vision and image processing. Two types of commonly
encountered noise are additive white Gaussian noise (AWGN) and multiplicative noise (MN). AWGN is often introduced in the optical imaging due to the thermal motion of electron in camera sensors or circuits, and MN (also called speckle) often appears in coherent imaging systems which are different from the usual optical imaging, e.g., synthetic aperture radar (SAR) or ultrasound imaging. By taking the logarithmic transform of the data, the MN or speckle noise model can be converted into a tractable additive noise [1]–[3]. Then many denoising techniques which are widely used for AWGN can also be applied for multiplicative noise removal [1], [2], [4]. In this article, we focus our attention on AWGN denoising problem.

In the last several years, the sparsity-inducing prior based on transformed frames has been playing a very important role in the development of effective image denoising methods. The fundamental idea behind is that the interest image or image patches can be sparsely represented by properly designed transformed frames. One such popular method is the wavelet tight frames [5], [6], which use fixed functions such as the linear or cubic B-spline as the generators and can be obtained by the unitary extension principle (UEP). Wavelet tight frames are widely used for image restoration tasks due to its strong ability of sparsely approximating piecewise smooth regions and preserving the jump discontinuities, and the low computational cost [7]–[11]. However, the structure of images varies greatly in practice, and a tight frame working well for one type of images may be not suitable for another. In order to properly represent the textures and tiny details including in the image, many approaches of learning an over-complete dictionary from the image [12]–[15] were developed in recent years. One of the representative work along this direction is the famous K-SVD method [14], which devises an alternating minimization algorithm to learn an over-complete dictionary from the noisy image patches, and use the learned dictionary to sparsely model and denoise images. These dictionary learning methods improve the quality of denoised images. However, the corresponding computational cost is too high due to the sparse coding of a large number of image patches is required during each iteration. Recently, the patch-dictionary learning methods were combined with another widely used nonlocal techniques [16], and generated various novel image restoration models [17]–[20]. These nonlocal sparsity representation methods were among the current state-of-the-art approaches in the quality of recovered images. However, the corresponding computational cost is really too high due to the computation of patch-similarity weights, patches group and sparsely coded for a large number of image patches. In fact, only the computation of similarity weights will spend much time.

Very recently, an adaptive tight frame construction scheme was proposed to overcome the drawback of the traditional wavelet tight frames [21]. Differently from the existing tight framelet methods, the
main idea of the new approach is to learn a tight frame from the input image. Through minimizing a problem with respect to the framelet filters and the canonical frame coefficient vectors, a data-driven tight frame can be efficiently learned by several iteration steps of simple singular value decomposition (SVD) and hard thresholding operator. Numerical experiments demonstrate that, with comparable performance in image denoising, the proposed data-driven tight frame construction scheme runs much faster than the general dictionary learning methods such as the classical K-SVD method. The proposed variational model in [21] is a non-convex model and the convergence of the corresponding alternating minimization algorithm is further investigated in the recent literature [22], which proves the sub-sequence convergence property of the algorithm. The data-adaptive tight framelet has also been directly applied to remove the multiplicative noise by simply using the logarithmic transform of the data [2].

In the above-mentioned data-driven tight frame construction algorithm, the framelet filters are updated by the simple SVD of a matrix with the size of $p^2 \times p^2$ at each iteration, where $p \times p$ is the size of the chosen filters. Therefore, the number of framelet filters learned from the input image is the same as the length of the support set of filters, which is also equal to $p^2$. However, in the implementation process of the proposed algorithm we observe that a majority of these learned framelet filters are unnecessary and even harmful to the denoised results due to the noise obviously influences the learned filters that correspond to the features that seldom appear in the input image. Based on this observed phenomena, we divide the matrix used for generating the framelet filters into two part, which correspond to the signal subspace and noise subspace respectively. Then we further propose an improved data-driven sparse representation method with much less number of filters learned by the bases of the signal subspace. Due to the removal of filters which are useless for the sparse representation of the image features, the computational cost is further reduced and the quality of denoising results is not damaged at the same time. The proposed data-driven framelet construction scheme is compared with the original algorithm with application to AWGN denoising problem.

The rest of this paper is organized as follows. In section [II] we briefly review the recently proposed data-driven tight frame construction algorithm. In section [III] in order to overcome the drawback of the previous work, we propose a new filters learning method, which excludes the inessential framelet filters. The convergence of the proposed algorithm is further investigated. In section [IV] the numerical examples on AWGN denoising are reported to compare the proposed algorithm with the original data-driven tight frame construction scheme.
II. DATA-DRIVEN TIGHT FRAME CONSTRUCTION SCHEME

In this section, we briefly introduce some preliminaries of wavelet tight frames, and then review the recently proposed data-driven tight frame construction scheme.

Starting with a finite set of generators \( \Psi = \{ \Psi_1, \cdots, \Psi_m \} \), one wavelet frame for \( L_2(\mathbb{R}) \) is just the affine system generated by the shifts and dilations of \( \Psi \), i.e., it is defined by

\[
X(\Psi) = \{ \Psi_{q,j,k}, \quad 1 \leq q \leq m, j \in \mathbb{Z}, k \in \mathbb{Z} \}
\]

where \( \Psi_{q,j,k} = 2^{j/2} \Psi_q(2^j \cdot -k) \). Such system \( X(\Psi) \) is called a tight frame of \( L_2(\mathbb{R}) \) if it satisfies

\[
f = \sum_{\psi \in \Psi} \langle f, \psi \rangle \psi \quad \text{for any } f \in L_2(\mathbb{R}).
\]

A tight frame has two associated operators: one is the analysis operator \( W \) defined by

\[
W : u \in L_2(\mathbb{R}) \rightarrow \{ \langle u, \Psi_{q,j,k} \rangle \},
\]

and the other is the adjoint operator \( W^T \) (also called the synthesis operator) defined by

\[
W^T : \{ \alpha_{q,j,k} \} \rightarrow \sum \alpha_{q,j,k} \Psi_{q,j,k} \in L_2(\mathbb{R}).
\]

The construction of wavelet tight frame is using the multi-resolution analysis (MRA). In the discrete setting, it can be generated by the shifts and dilations of a set of masks or filters \( \{ h_1, \cdots, h_m \} \) associated with the corresponding generators \( \Psi \). The construction of these filters is based on the unitary extension principle (UEP). For more details refer to [5], [6]. The widely used framelet in image recovery is the linear B-spline framelet, which is constructed by three filters as follows:

\[
h_0 = \frac{1}{4}[1, 2, 1]; \quad h_1 = \frac{\sqrt{2}}{4}[1, 0, -1]; \quad h_2 = \frac{1}{4}[-1, 2, -1].
\]

Once given the 1-dimensional wavelet tight frame, the framelets for \( L_2(\mathbb{R}^2) \) can be easily constructed by taking tensor products of 1-dimensional framelets.

However, in the tight frame systems introduced above, the filters are fixed and unable to be adaptively adjusted according to the input image. Therefore, Cai et.al [21] further proposed a variational model to learn a tight frame system from the image itself. The tight frames considered there are single-level un-decimal discrete wavelet systems generated by the data-driven filters \( \{ h_1, \cdots, h_m \} \). For any filter \( h \), let \( S_h \) denote its associated convolution operator. Then the analysis operator \( W \) and the synthesis operator \( W^T \) can be explicitly derived as

\[
W(h_1, \cdots, h_m) = \left[ S_{h_1}^T, S_{h_2}^T, \cdots, S_{h_m}^T \right]^T; \quad W^T(h_1, \cdots, h_m) = [S_{h_1}, S_{h_2}, \cdots, S_{h_m}] \tag{1}
\]
where \( h(\cdot) \) is obtained via transposing matrix \( h \) first and then reversing the elements of the transposition.

The proposed model in [21] for constructing the data-driven tight frame can be briefly summarized as follows:

\[
\min_{v, \{h_i\}_{i=1}^m} \| v - W(h_1, \cdots, h_m)g \|_2^2 + \lambda^2_0 \| v \|_0, \quad \text{s.t. } W^TW = I (2)
\]

where \( g \) denotes the noisy image, and \( v \) is the coefficient vector that sparsely approximates the tight frame coefficient \( Wg \). For the general case of framelet filters, the minimization problem (2) is very difficult to solve. Therefore, the authors in [21] consider a special case of \( m = p^2 \) and each \( h_i \) be a real-valued filter with support on \( \mathbb{N}^2 \cap [1, p]^2 \). For each \( h_i \), let \( \vec{h_i} \) be its column vector form by concatenating all its columns. Define \( H = [\vec{h_1}, \vec{h_2}, \cdots, \vec{h_{p^2}}] \). It has been proved that the constraint \( W^TW = I \) exists as long as \( H^TH = \frac{1}{p^2}I \). The interested reader can refer to Proposition 3 of [21] for details. Therefore, the minimization problem (2) can simply be reformulated as

\[
\min_{v, \{h_i\}_{i=1}^m} \| v - W(H)g \|_2^2 + \lambda^2_0 \| v \|_0, \quad \text{s.t. } H^TH = \frac{1}{p^2}I. (3)
\]

The problem (3) can be rewritten in the matrix form. Let \( \{\vec{g}_i\}_{i=1}^N \in \mathbb{R}^{p^2} \) be the set of image patches of size \( p \times p \) densely extracted from the image \( g \). For each patch \( \vec{g}_i \), let \( \vec{v}_i \) denote the vector corresponding to \( H^T\vec{g}_i \in \mathbb{R}^{p^2} \). Define

\[
\begin{align*}
G &= \frac{1}{p} (\vec{g}_1, \vec{g}_2, \cdots, \vec{g}_N) \in \mathbb{R}^{p^2 \times N}, \\
V &= (\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_N) \in \mathbb{R}^{p^2 \times N}, \\
D &= p (\vec{h_1}, \vec{h_2}, \cdots, \vec{h_{p^2}}) \in \mathbb{R}^{p^2 \times p^2}.
\end{align*}
\]

Then problem (3) is equal to

\[
\min_{V,D} \| V - D^TG \|_F^2 + \lambda^2 \| V \|_0, \quad \text{s.t. } D^TD = I. (5)
\]

where \( \lambda \) represents some fixed regularization parameter. In literature [21], this problem is solved via an alternating minimization scheme between the coefficients \( V \) and the learning filters \( D \). More specifically, given the current iteration value \( D_k \) of the variable \( D \), the next iteration update for \( V \) is via solving the following minimization problem

\[
\min_{V} \| V - D_k^TG \|_F^2 + \lambda^2 \| V \|_0. (6)
\]
It is easy to deduce that the optimal value $V$ of the minimization problem (6) can be simply obtained by the hard thresholding operator. In what follows, fixed the value of $V$ as $V_{k+1}$, the next iteration update for $D$ is given by the minimization problem

$$\min_D \|V_{k+1} - D^T G\|_F^2, \quad \text{s.t.} \quad D^T D = I.$$  \hfill (7)

Let $U_k \Sigma_k X_k^T$ be the SVD of $GV_{k+1}^T \in \mathbb{R}^{p^2 \times p^2}$ such that $GV_{k+1}^T = U_k \Sigma_k X_k^T$. It has been shown in [21] that the minimization problem (7) has a closed solution denoted by

$$D_{k+1} = U_k X_k^T.$$  \hfill (8)

It is observed that the filters set $D$ associated with the framelet can be obtained by a simple SVD of a matrix of small size $p^2 \times p^2$.

### III. Improved Data-Driven Sparse Representation Method

#### A. Motivation and model description

In the above-mentioned algorithm, the matrix $D$ associated with the framelet filters is updated by the iterative formula (8), where $U_k$ and $X_k$ are given by the SVD of $GV_{k+1}^T$. Assume that $\Sigma_k = \text{diag}\{r_i\}$, where $\{r_i\}_{i=1}^{p^2}$ denotes the singular value of $GV_{k+1}^T$ in a descending order. Let $U_k = \{u_1^k, u_2^k, \ldots, u_{p^2}^k\}$ and $X_k = \{x_1^k, x_2^k, \ldots, x_{p^2}^k\}$. Then $GV_{k+1}^T$ can be reformulated as

$$GV_{k+1}^T = \sum_{i=1}^{p^2} r_i u_i^k (x_i^k)^T = \underbrace{\sum_{i=1}^{s} r_i u_i^k (x_i^k)^T}_\text{signal subspace} + \underbrace{\sum_{i=s+1}^{p^2} r_i u_i^k (x_i^k)^T}_\text{noise subspace}. \hfill (9)$$

In the denoising problem, the input image $g$ used for generating the matrix $G$ is contaminated by noise, thus so is the matrix $GV_{k+1}^T$. Therefore, we can divide the whole space spanned by the bases $\{u_i^k(x_i^k)^T\}_{i=1}^{p^2}$ into two orthogonal subspace: the signal subspace which is spanned by the bases corresponding to the $s$ ($s < p^2$) largest singular values; and the noise subspace which is spanned by the bases corresponding to the rest singular values. Then in the formula (9) the matrix $GV_{k+1}^T$ is divided into two part—one is the signal component obtained by projection on the signal subspace, the other is the noise component obtained by projection on the noise subspace.

Rewrite $D_{k+1} = \sum_{i=1}^{s} u_i^k (x_i^k)^T + \sum_{i=s+1}^{p^2} u_i^k (x_i^k)^T$. the last part of $\sum_{i=s+1}^{p^2} u_i^k (x_i^k)^T$ can be regarded as the filters component mainly learned from the noise component of the input image, and hence maybe useless.
and even harmful to the final denoised results. Therefore, it should be removed in the filters learning process.

To illustrate this conclusion more clearly, we use the Barbara image of size $256 \times 256$ contaminated by zero-mean AWGN of $\sigma = 25$ as an example, where $\sigma$ denotes the standard deviation of the noise. Figure 1(c) shows the curve of the singular values of $G V_T^T K (K = 25)$. It is observed that the singular values $r_i$ decrease very fast with the increase of the index $i$. This phenomenon implies that the framelet filters corresponding to the last several singular values of $G V_T^T$ will be influenced by the noise more obviously compared with those corresponding to the larger singular values. Figure 1(d) presents the obtained adaptive tight frame filters generated by $\sum_{i=1}^{p^2} u_k^i(x_k^i)^T$. It is found that many filters included in the figure are likely to be contaminated by the noise, e.g., the second to seventh filters in the first row, and the fifth to seventh filters in the second row of Figure 1(c). This is due to the fact that these filters are generated by the bases corresponding to the last several singular values of $G V_T^T$ and hence influenced by the noise obviously. Figures 1(e) and 1(f) also show the frame filters generated by $\sum_{i=1}^{s} u_k^i(x_k^i)^T$ and $\sum_{i=s+1}^{p^2} u_k^i(x_k^i)^T$, where we choose $s = 30$. It is clearly observed that the filters contained in the noisy subspace have been severely influenced by the noise, and hence are not suitable for the sparse representation of the image.

Motivated by the above analysis, we propose a new strategy to remove the useless framelet filters and develop an improved data-driven sparse representation method. Assume that only $s (s < p^2)$ filters including in the filters group $H$ are suitable for the sparse representation of the noisy image. Without loss of generality, we group filters $H$ into two classes: $H_1 = [h_1, h_2, \cdots, h_s]$ and $H_2 = [h_{s+1}, h_{s+2}, \cdots, h_{p^2}]$, and assume that $H_1$ is used for the sparse modeling of the noisy image. Then similarly to (3), we can further propose a variational model for adaptive filters learning as follows

$$\min_{v, H_1, H_2} \| v - W(H_1)g \|_2^2 + \| W(H_2)g \|_2^2 + \lambda_0^2 \| v \|_0, \quad \text{s.t.} \quad H^T H = \frac{1}{p^2} I. \quad (10)$$

In the proposed model, it is only demanded that the frame coefficients generated by the filters group $H_1$ are sparse. Besides, an extra regularization term $\| W(H_2)g \|_2^2$ is introduced to guarantee the frame coefficients produced by group $H_2$ are small enough, and also for the convenience of computation. In what follows, we further reformulate the minimization problem (10) in the matrix form. Define
Then problem (10) can be rewritten as

$$\min_{D,V} \|V - D^T G\|_F^2 + \|D^T_2 G\|_F^2 + \lambda^2 \|V\|_0, \quad \text{s.t.} \quad D^T D = I.$$  \hspace{1cm} (12)

The established model can be solved by the alternating minimization scheme between $V$ and $D$. Specifically, given the current estimate $(V_k, D_k)$, the next iteration updates its value via the following scheme:
\begin{equation}
\begin{aligned}
V_{k+1} &= \arg \min_{V \in \mathbb{R}^{s \times N}} \|V - (D_1)^T_k G\|_F^2 + \lambda^2 \|V\|_0, \\
D_{k+1} &= \arg \min_{D \in \mathbb{R}^{p_2 \times p_2}} \|V_{k+1} - D_1^T G\|_F^2 + \|D_2^T G\|_F^2, \quad \text{s.t.} \quad D^T D = I.
\end{aligned}
\end{equation}

The first sub-problem of (13) has a closed solution as follows:

\begin{equation}
V_{k+1} = \mathcal{H}_\lambda((D_1)^T_k G)
\end{equation}

where \( \mathcal{H}_\lambda : \mathbb{R}^{s \times N} \rightarrow \mathbb{R}^{s \times N} \) is the hard thresholding operator defined by

\[
[H_\lambda(U)]_{i,j} = \begin{cases} U_{i,j}, & \text{if } |U_{i,j}| > \lambda, \\ 0, & \text{otherwise}. \end{cases}
\]

The objective function of the second sub-problem of (13) can be reformulated as

\begin{equation}
\|V_{k+1} - D_1^T G\|_F^2 + \|D_2^T G\|_F^2 = \text{Tr}((V_{k+1} - D_1^T G)(V_{k+1} - D_1^T G)^T) + \text{Tr}(G^T D_2 D_2^T G)
\end{equation}

\begin{equation}
= \text{Tr}(V_{k+1}^T V_{k+1}) + \text{Tr}(G^T D_2 D_2^T G) - 2\text{Tr}(D_1^T G V_{k+1}^T)
\end{equation}

\begin{equation}
= \text{Tr}(V_{k+1}^T V_{k+1}) + \text{Tr}(G^T G) - 2\text{Tr}(D_1^T G V_{k+1}^T)
\end{equation}

Note that the third equality of (16) is from the fact that \( DD^T = I \), which is derived directly from \( D^T D = I \). Therefore, the optimal value \( (D_1)_{k+1} \) is the solution of the following problem

\begin{equation}
\max_{D_1} \text{Tr}(D_1^T G V_{k+1}^T) \quad \text{s.t.} \quad D_1^T D_1 = I.
\end{equation}

Suppose that the SVD of \( GV_{k+1}^T \in \mathbb{R}^{p_2 \times s} \) is \( \tilde{U}_{k+1} \tilde{\Sigma}_{k+1} \tilde{X}_{k+1}^T \), where \( \tilde{U}_{k+1} \in \mathbb{R}^{p_2 \times s} \), \( \tilde{\Sigma}_{k+1} \in \mathbb{R}^{s \times s} \), and \( \tilde{X}_{k+1} \in \mathbb{R}^{s \times s} \). Then based on Theorem 4 in [21] we know that the minimization problem (17) has a closed solution

\begin{equation}
(D_1)_{k+1} = \tilde{U}_{k+1} \tilde{X}_{k+1}^T.
\end{equation}

The value of \( (D_2)_{k+1} \) should satisfies that \( D_1^T_{k+1} D_{k+1} = I \). In other words, it should be the orthocomplement of \( (D_1)_{k+1} \). Due to it represents the useless filters and is independent with the update of \( V \), the corresponding computation can be neglected here. Following the above analysis, we obtain the improved data-driven sparse representation method, which is summarized in Algorithm 1. After the learned filters set \( (D_1)_K \) is obtained, the denoised image can be given by
\[ g_K = (W((D_1)_k))^T(H_{\tilde{\lambda}}(W((D_1)_k)g)). \]  

where \( \tilde{\lambda} \) is the threshold determined by the noise level.

**Algorithm 1** Construction of the data-driven filters for sparse representation

**Input:** the noisy \( g \).

**Output:** data-driven filters set \( D_1 \) defined by \( p(\vec{h}_1, \vec{h}_2, \cdots, \vec{h}_s) \in \mathbb{R}^{p^2 \times s} \).

**Initialization:** set the number of filters \( s \), and initial filter matrix \( D_0 \); construct the patch matrix \( G \) defined by (11).

**Iteration:** For \( k = 0, 1, \cdots, K - 1 \)

(i) update \( V \):

\[ V_{k+1} = H_{\tilde{\lambda}}((D_1)_k^T G); \]

(ii) update \( D_1 \):

compute the SVD decomposition of \( GV_{k+1}^T = \tilde{U}_{k+1} \tilde{\Sigma}_{k+1} \tilde{X}_{k+1}^T \); and update \( D_1 \) as \( (D_1)_{k+1} = \tilde{U}_{k+1} \tilde{X}_{k+1}^T \).

**B. Further analysis of the proposed model**

In this section, we further analyze the convergence of the proposed algorithm. In recent years, the convergence of the alternating proximal minimization algorithm for non-convex minimization problems has been widely investigated [23], [24]. One of the widely used tool is the Kurdyka-Lojasiewicz (K-L) property, see [22], [23], [25] for more details. In literature [22], the authors used the results presented in [23] to further investigate the convergence of the data-driven tight frame construction scheme introduced in section II The convergence of the proposed algorithm can be proved very similarly.

Denote the convex set

\[ \Omega_{\tilde{V}} = \{ v_{i,j} \mid v_{i,j} = 0 \text{ for any } i > s \} \]

for any \( \tilde{V} \in \mathbb{R}^{p^2 \times N} \). Then the proposed model (12) can be reformulated as

\[ \min_{D, V \in \Omega_{\tilde{V}}} \| \tilde{V} - D^T G \|^2_F + \lambda^2 \| V \|_0, \text{ s.t. } D^T D = I. \]  

(20)

Define

\[ f(\tilde{V}) = \lambda^2 \| \tilde{V} \|_0 + \iota_{\Omega_{\tilde{V}}}(\tilde{V}), \quad Q(\tilde{V}, D) = \| \tilde{V} - D^T G \|^2_F, \quad h(D) = \iota_{\Omega_0}(D) \]  

(21)
where \( \iota_\Omega \) denotes the indicative function of the set \( \Omega \), i.e., \( \iota_\Omega(A) = 0 \) if \( A \in \Omega \), and \(+\infty\) otherwise. Then, the minimization problem (20) can be rewritten as

\[
\min_{D,\tilde{V}} \mathcal{L}(\tilde{V}, D) = f(\tilde{V}) + h(D) + Q(\tilde{V}, D).
\] (22)

Similarly to the proof in literature [22], we can easily derive the following convergence result. Due to the proof requires only minor changes to that presented in [22], we omit it here due to limited space.

**Theorem 1:** (The convergence of Algorithm 1) The sequence \( \{D^k, V^k\} \) generated by Algorithm 1 has at least one limit point. Let \((D^*, V^*)\) be any limit point of the sequence \(\{D^k, V^k\}\). Denote \(\tilde{V}^* = (V^*; 0) \in \mathbb{R}^{p \times N}\). Then \((D^*, \tilde{V}^*)\) is a stationary point of (22).

However, the above conclusion only illustrates the sub-sequence convergence property of the proposed algorithm. The sequence convergence property can be guaranteed by coupling this alternating minimization method with a proximal term. See [22]–[24] for more details. In this setting, the proposed iteration scheme can be modified as

\[
\begin{align*}
\tilde{V}_{k+1} &= \text{arg} \min_{\tilde{V}} \mathcal{L}(\tilde{V}, D^k) + \lambda_k \|\tilde{V} - \tilde{V}_k\|^2_F, \\
D_{k+1} &= \text{arg} \min_D \mathcal{L}(\tilde{V}_{k+1}, D) + \mu_k \|D - D_k\|^2_F \\
\end{align*}
\] (23)

where \(\lambda_k, \mu_k \in (c_1, c_2)\) and \(c_1, c_2\) are two positive constants. Consequently, the iteration scheme of Algorithm 1 can be modified as

\[
\begin{align*}
V_{k+1} &= \mathcal{H}_{\lambda/\sqrt{1 + \lambda_k}} \left( \frac{\langle D_1 \rangle G + \lambda_k V_k}{1 + \lambda_k} \right), \\
(D_1)_{k+1} &= \tilde{U}_{k+1} \tilde{\Sigma}_{k+1} \tilde{X}_{k+1}^T \\
\end{align*}
\] (24)

where \(\tilde{U}_{k+1}\) and \(\tilde{X}_{k+1}\) are given by the SVD of \(GV_{k+1}^T + \mu_k (D_1)_k = \tilde{U}_{k+1} \tilde{\Sigma}_{k+1} \tilde{X}_{k+1}^T\). Then similarly to the results shown in [22], we can easily deduce the following conclusion.

**Theorem 2:** The sequence \( \{D^k, V^k\} \) generated by (23) converges to a stationary point of (22).

In fact, the lack of sequence convergence is not crucial for the application of image denoising due to the reason that the results we are seeking for are not the frame coefficients but the image synthesized from the coefficients. Numerical experiments in [22] also demonstrate that the algorithms with and without the proximal term shown in (23) have almost the same denoising performance in terms of PSNR values. The conclusion also exists for the proposed algorithm. Therefore, we choose \(\lambda_k = \mu_k = 0\) in the following experiments.
IV. NUMERICAL EXPERIMENTS

In this section, we compare the proposed algorithm (shown as Algorithm 1) with the original data-driven tight frame construction scheme [21] in the field of AWGN denoising. Both the quality of the recovery images and the computational costs of these algorithms are compared.

The codes of Algorithm 1 and the original data-driven tight frame method are written entirely in Matlab. All the numerical examples are implemented under Windows XP and MATLAB 2009 running on a laptop with an Intel Core i5 CPU (2.8 GHz) and 8 GB Memory. In the following experiments, six standard nature images with size of $512 \times 512$ (see Figure 2), which consist of complex components in different scales and with different patterns, are used for our test.

A. The analysis of the proposed algorithm

First of all, we simply analyze the computational costs of both the original data-driven tight frame construction scheme [21] and the proposed algorithm (shown as Algorithm 1). It is observed that each iteration of both algorithms includes two stages: hard thresholding and tight frame update. For the original algorithm in [21], the size of matrix $D^T G$ used for hard thresholding operator is $p^2 \times N$, and the tight

1http://www.math.nus.edu.sg/ matjh/download/data_driven_tight_frames/data_driven_tight_frame_ver1.0.zip
frame update is done via the SVD of the matrix $GV^T$ with the size of $p^2 \times p^2$. For Algorithm 1, the size of the corresponding matrix used for hard thresholding operator is $s \times N$, and the SVD used for the tight frame update is related to the matrix with the size of $s \times p^2$. Therefore, the whole computational cost can be reduced obviously by choosing a small $s$. Besides, the computational burden of the analysis operator and the synthesis operator generated by $D_1 \in \mathbb{R}^{p^2 \times s}$, which corresponds to $s$ times of convolution operator, is also small than those produced by $D \in \mathbb{R}^{p^2 \times p^2}$. Hence, the computational cost of generating the denoised image (see the formula (19)) is also smaller than that of the original data-driven tight frame method.

In what follows, the initialization of the data-driven filters set $D_1$ and the selection of the parameter $s$ are further discussed. The initialization of tight frames has been investigated in literature [21], and one suitable selection is to use DCT as the initial guess. Denote $D = \{d_1, d_2, \cdots, d_{p^2}\}$ as the tight frame generated by $p \times p$ DCT. We can simply generate the initial $D_1$ by randomly choosing $s$ elements from $D$. On the other hand, inspired by the discussion in section III-A (refer to (9) and Figure 1), we can use the signal component $S = \sum_{i=1}^{s} u_i^T(x_k)$ to obtain the initial $D_1$. Specifically, we choose $s$ elements from $S$ that satisfy $\|e_i\|_2 > \epsilon$, where $e_i$ denote the elements of $S$, and $\epsilon$ is a small constant such as $10^{-5}$ to guarantee the chosen elements are not too small and may represent the main feature of the image. The matrix $S$ can be generated by one or two iteration steps (In our experiments we choose two iteration steps) of the original data-driven tight frame construction scheme.

It is noted that one important parameter of the proposed algorithm is the filters number $s$. In the next, three images, Barbara, Lena and Boat, are used for testing the performance of Algorithm 1 with different values of $s$ and initialization of filters set $D_1$. The size $p$ is chosen to be 8, which is a suitable selection by considering both the recovery quality and implementation time. For the thresholding parameters $\lambda$ and $\tilde{\lambda}$, we use the default setting in the recent published code of literature [21], i.e., fix $\lambda = 3.4\sigma$ and $\tilde{\lambda} = 2.7\sigma$, which is the suitable selection through many trials. The maximum iteration number is set to 25. The performance is quantitatively measured by means of PSNR, which is expressed as

$$\text{PSNR}(u, u^*) = -20 \log \left( \frac{\|u - u^*\|_2}{255N} \right)$$

where $u$ and $u^*$ denote the original and restored images, respectively, and $N$ is the total number of pixels in the image $u$.

Table I lists the PSNR values and CPU time for various selection of $s$ values and initialization of $D_1$. Here $s = 10, \cdots, 50$ represent that the corresponding initialization of $D_1$ is obtained by the signal
component $S$, and $DCT$, $s = 10, 20$ demonstrate that the initialization is given by the DCT. It is observed that $s = 20$ or $30$ is a suitable selection for Algorithm 1, which means that $s$ can be chosen as $p^2/3$ or $p^2/2$ approximately. Besides, the performance of the algorithm with the initialization of $D_1$ from $S$ overall outperforms that using the DCT as the initialization. In the following experiments, we use the better initialization strategy for $D_1$.

### TABLE I

**The comparison of the performance of Algorithm 1 with different $s$ and initialization**

| Image   | noise index | $s = 10$ | $s = 20$ | $s = 30$ | $s = 40$ | $s = 50$ | $DCT, s = 20$ | $DCT, s = 30$ |
|---------|-------------|----------|----------|----------|----------|----------|----------------|----------------|
| **Barbara** | $\sigma = 30$ | PSNR 26.16 | 27.84 28.57 | 28.56 28.53 | 27.83 28.54 |
|         | Time 1.10 1.62 2.21 2.79 3.39 | 1.59 2.15 |
|         | $\sigma = 40$ | PSNR 25.21 26.54 27.15 | 27.12 27.07 | 26.69 27.08 |
|         | Time 1.06 1.68 2.16 2.73 3.43 | 1.57 2.15 |
|         | $\sigma = 50$ | PSNR 24.94 25.64 26.00 | 25.89 25.88 | 25.54 25.80 |
|         | Time 1.09 1.63 2.19 2.73 3.26 | 1.73 2.11 |
| **Lena**  | $\sigma = 30$ | PSNR 30.41 30.58 30.59 | 30.51 30.44 | 30.52 30.51 |
|         | Time 1.09 1.68 2.17 2.72 3.29 | 1.58 2.14 |
|         | $\sigma = 40$ | PSNR 29.18 29.23 29.18 | 29.12 29.02 | 29.08 29.14 |
|         | Time 1.07 1.64 2.19 2.72 3.20 | 1.56 2.13 |
|         | $\sigma = 50$ | PSNR 28.15 28.16 28.07 | 27.99 27.86 | 28.06 28.04 |
|         | Time 1.10 1.63 2.20 2.74 3.29 | 1.57 2.12 |
| **Boat**  | $\sigma = 30$ | PSNR 28.01 28.48 28.53 | 28.52 28.47 | 28.51 28.48 |
|         | Time 1.07 1.63 2.18 2.72 3.31 | 1.57 2.11 |
|         | $\sigma = 40$ | PSNR 26.98 27.21 27.21 | 27.20 27.15 | 27.15 27.14 |
|         | Time 1.06 1.65 2.19 2.70 3.26 | 1.72 2.12 |
|         | $\sigma = 50$ | PSNR 26.11 26.24 26.17 | 26.15 26.08 | 26.14 26.16 |
|         | Time 1.08 1.64 2.15 2.73 3.27 | 1.56 2.13 |

**B. Denoising performance evaluation on additive white Gaussian noise**

In this section, we report the experimental results, comparing the proposed algorithm with the original data-driven tight frame construction scheme [21], on the denoising of additive Gaussian noise. Six natural images with size of $512 \times 512$ are used for our test. Considering both the recovery quality and computational time, we set the maximum iteration number to $25$ for both algorithms through all experiments. In this experiment, we consider the filters with size of $8 \times 8$ and $16 \times 16$ respectively. The selection of the thresholding parameters $\lambda$ and $\tilde{\lambda}$ is the same as that mentioned above. Table II lists the PSNR values and CPU time of different algorithms. Here the original data-driven tight frame
construction scheme [21] is abbreviated as “DDTF”, and “DDTF; 8” represents the DDTF method with 
\( p = 8 \). Similarly, “Alg1(20);8” denotes Algorithm 1 with \( p = 8 \) and \( s = 20 \), where the number in the 
bracket represents the value of \( s \), i.e., the number of chosen filters. Two values of \( s \), which approximately 
equal to \( p^2/3 \) or \( p^2/2 \), are tested here.

From the results in Table II, we observe that the proposed algorithm overall outperforms the DDTF 
method in both the recovery quality and computational time, especially for the higher noise level. This 
is due to more filters learned from the SVD of the matrix \( GVT \) are influenced by the several noise. 
As a result, the coefficients generated by these filters may not be sparse, and removing these filters can 
 improve the recovery quality and reduce the computational cost meanwhile. In general, comparing the 
indexes in Table II, we can easily draw the following conclusions.

1) The DDTF method with \( p = 8 \) takes about \( 4.0 \sim 4.1 \) s, Algorithm 1 with \( p = 8 \) and \( s = 20 \) takes 
about \( 1.6 \sim 1.7 \) s, and Algorithm 1 with \( p = 8 \) and \( s = 30 \) takes about \( 2.1 \sim 2.2 \) s. The computational 
time of the corresponding algorithms increases more than ten times due to the fact that both the size of 
the support set of filters and the number of filters increase four times.

2) The small \( s \) is suitable for images with less textures and higher noise level, and the improvement of 
recovery quality becomes more obvious while noise level increases. For the case of \( p = 8 \), it is observed 
that Algorithm 1 with \( s = 20 \) obtains the best PSNRs for almost all the condition that \( \sigma \geq 40 \) (except 
the Barbara image which is rich in terms of textures).

3) These algorithms with \( p = 16 \) outperform those with \( p = 8 \) for the images Barbara, Lena and 
Fingerprint, especially when the noise is large. Maybe it is the reason that these images have more 
complex texture regions.

Figure 3 shows the denosing results of different algorithms, and the corresponding learned filters are 
presented in Figure 4. As observed, part of these filters generated by the DDTF method are contaminated 
by noise terribly due to the influence of high noise. However, these filters are not included in the filters 
generated by our method. An visual observation of the results for the Lena image can also be obtained 
in Figure 5.

V. CONCLUSION

In this paper, inspired by the fact that part of the data-driven tight frame filters learned by the recently 
proposed method [21] are influenced by the noise severely, we propose an improved data-driven filters 
learning method. In our method, we divide the matrix used for generating the tight frame filters into two 
part, corresponding to the so called signal subspace and noise subspace. Then only the bases spanning the
TABLE II
THE COMPARISON OF THE PERFORMANCE OF ALGORITHM 1 AND THE ORIGINAL ALGORITHM [22]

| Image   | Algorithm | PSNR | Time (s) |
|---------|-----------|------|----------|
| Barbara | DDTF [21];8 | 30.60 | 4.04 |
|         | Alg1(20);8 | 29.34 | 1.63 |
|         | Alg1(30);8 | 30.52 | 2.18 |
| Lena    | DDTF [21];16 | 31.00 | 18.60 |
|         | Alg1(80);16 | 29.91 | 2.20 |
|         | Alg1(120);16 | 30.85 | 2.72 |
| Fingerprint | DDTF [21];8 | 28.36 | 4.07 |
|         | Alg1(20);8 | 28.42 | 1.71 |
|         | Alg1(30);8 | 28.44 | 2.22 |
| Boat    | DDTF [21];16 | 28.36 | 4.17 |
|         | Alg1(80);16 | 28.37 | 19.09 |
|         | Alg1(120);16 | 28.41 | 27.18 |

σ values: 20, 30, 40, 50, 60, 70
Fig. 3. Visual comparison of denoising results. (a) Noisy image with $\sigma = 40$, (b) result by DDTF method with $p = 8$, (c) result by Alg.1 with $p = 8$ and $s = 30$, (d) result by DDTF method with $p = 16$, (e) result by Alg.1 with $p = 16$ and $s = 80$, (f) result by Alg.1 with $p = 16$ and $s = 120$.

Fig. 4. The data-driven tight frame filters. (a) result by DDTF method with $p = 8$, (b) result by Alg.1 with $p = 8$ and $s = 20$, (c) result by Alg.1 with $p = 8$ and $s = 30$, (d) result by DDTF method with $p = 16$, (e) result by Alg.1 with $p = 16$ and $s = 80$, (f) result by Alg.1 with $p = 16$ and $s = 120$. 
Fig. 5. Visual comparison of denoising results. (a) Noisy image with $\sigma = 30$, (b) result by DDTF method with $p = 8$, (c) result by Alg.1 with $p = 8$ and $s = 20$, (d) result by Alg.1 with $p = 8$ and $s = 30$, (e) result by DDTF method with $p = 16$, (f) result by Alg.1 with $p = 16$ and $s = 80$.

signal subspace are used for constructing the learned filters. This means that the filters contaminated by the noise severely are excluded for the sparsity representation of images. Numerical experiments demonstrate that our method overall outperforms the original data-driven tight frame construction scheme.

REFERENCES

[1] Durand S, Fadili J, Nikolova M. Multiplicative noise removal using L1 fidelity on frame coefficients. Journal of Mathematical Imaging and Vision, 2010, 36(3): 201-226.

[2] Feng W, Lei H. SAR Image Despeckling Using Data-Driven Tight Frame. Geoscience and Remote Sensing Letters, IEEE, 2014, 11(9): 1455-1459.

[3] Chen D Q, Zhou Y. Multiplicative Denoising Based on Linearized Alternating Direction Method Using Discrepancy Function Constraint. Journal of Scientific Computing. 2014, 60(3): 483-504.

[4] Huang Y M, Moisan L, Ng M K, et al. Multiplicative noise removal via a learned dictionary. Image Processing, IEEE Transactions on, 2012, 21(11): 4534-4543.

[5] Daubechies I, Han B, Ron A, et al. Framelets: MRA-based constructions of wavelet frames. Applied and computational harmonic analysis, 2003, 14(1): 1-46.

[6] Dong B, Shen Z. MRA based wavelet frames and applications. IAS Lecture Notes Series, Summer Program on The Mathematics of Image Processing, Park City Mathematics Institute, 2010.

[7] Cai J F, Chan R H, Shen Z. A framelet-based image inpainting algorithm. Applied and Computational Harmonic Analysis, 2008, 24(2): 131-149.
[8] Cai J F, Osher S, Shen Z. Linearized Bregman iterations for frame-based image deblurring. SIAM Journal on Imaging Sciences, 2009, 2(1): 226-252.
[9] Cai J F, Dong B, Osher S, et al. Image restoration: Total variation, wavelet frames, and beyond. Journal of the American Mathematical Society, 2012, 25(4): 1033-1089.
[10] Cai J F, Ji H, Liu C, et al. Framelet-based blind motion deblurring from a single image. Image Processing, IEEE Transactions on, 2012, 21(2): 562-572.
[11] Chen D Q. Regularized Generalized Inverse Accelerating Linearized Alternating Minimization Algorithm for Frame-Based Poissonian Image Deblurring. SIAM Journal on Imaging Sciences, 2014, 7(2): 716-739.
[12] Kreutz-Delgado K, Murray J F, Rao B D, et al. Dictionary learning algorithms for sparse representation. Neural computation, 2003, 15(2): 349-396.
[13] Elad M. Sparse and redundant representations: from theory to applications in signal and image processing. Springer Science Business Media, 2010.
[14] Elad M, Aharon M. Image denoising via sparse and redundant representations over learned dictionaries. Image Processing, IEEE Transactions on, 2006, 15(12): 3736-3745.
[15] Mairal J, Sapiro G, Elad M. Learning Multiscale Sparse Representations for Image and Video Restoration. Multiscale Modeling Simulation, 2008, 7(1): 214-241.
[16] Buades A, Coll B, Morel J M. A review of image denoising algorithms, with a new one. Multiscale Modeling and Simulation, 2005, 4(2): 490-530.
[17] Mairal J, Bach F, Ponce J, et al. Non-local sparse models for image restoration. IEEE International Conference on Computer Vision (ICCV), 2009, pp. 2272-2279.
[18] Zhang L, Dong W, Zhang D, et al. Two-stage image denoising by principal component analysis with local pixel grouping. Pattern Recognition, 2010, 43(4): 1531-1549.
[19] Dong W, Li X, Zhang D, et al. Sparsity-based image denoising via dictionary learning and structural clustering. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 457-464.
[20] Dong W, Zhang L, Shi G, et al. Nonlocally centralized sparse representation for image restoration. IEEE Transactions on Image Processing, 2013, 22(4): 1620-1630.
[21] Cai J F, Ji H, Shen Z, et al. Data-driven tight frame construction and image denoising. Applied and Computational Harmonic Analysis, 2014, 37(1): 89-105.
[22] Bao C, Ji H, Shen Z. Convergence analysis for iterative data-driven tight frame construction scheme. Applied and Computational Harmonic Analysis, in press, 2015.
[23] Attouch H, Bolte J, Redont P, et al. Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the Kurdyka-Lojasiewicz inequality. Mathematics of Operations Research, 2010, 35(2): 438-457.
[24] Bolte J, Sabach S, Teboulle M. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Mathematical Programming, 2014, 146(1-2): 459-494.
[25] Bolte J, Daniilidis A, Lewis A. The Lojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. SIAM Journal on Optimization, 2007, 17(4): 1205-1223.