Convergent Convolutional Dictionary Learning Using Adaptive Contrast Enhancement (CDL-ACE): Application of CDL to Image Denoising

Il Yong Chun* and Jeffrey A. Fessler†
Department of Electrical Engineering and Computer Science, The University of Michigan
Ann Arbor, MI 48019-2122 USA
Email: *iychun@umich.edu, †fessler@umich.edu

***Abstract*—*Convolutional dictionary learning* (CDL) has great potential to "learn" rich sparse representations from training datasets, by training translation-invariant filters. However, the performance of applying learned filters from CDL to inverse problems has not yet been fully maximized because training data preprocessing in training stage is not fully compensated in testing stage. We propose *CDL using Adaptive Contrast Enhancement* (CDL-ACE) that additionally models the preprocessing in CDL, and image denoising model using learned filters from CDL-ACE. For CDL-ACE, we apply a practically feasible and convergent *Block Proximal Gradient method using Majorizer* (BPG-M) with a momentum coefficient formula and an adaptive restarting rule. Numerical experiments show that, for strong additive white Gaussian noise, the proposed image denoiser using learned filters by CDL outperforms existing image denoising methods using Wiener filtering and total variation; and learned filters by CDL-ACE further improves the denoiser.**

I. Introduction

Convolutional dictionary learning (CDL; or convolutional sparse coding [1]–[5]) can overcome the fundamental problems of patch-based dictionary learning [6], [7]: *1)* translation-variant dictionaries and *2)* highly redundant sparse representations [5], [8]. In addition, CDL is closely related to (deep) convolutional neural networks (CNN) [1], [8], [9]. Learned filters by CDL have been successfully applied to various computer vision problems (see references within [4], [5]); however, applying them to inverse problems is not straightforward due to *model mismatch* between training and testing stages. The main reason for the model mismatch is that CDL learns features from preprocessed training datasets (by, for example, local contrast enhancement, high-pass filtering, and subtracting the mean [1], [2], [4]); however, the preprocessing techniques are not directly suitable for solving inverse problems [10].

We propose *1) CDL using Adaptive Contrast Enhancement* (CDL-ACE) which integrates the preprocessing principles into the CDL formulation and *2)* image denoising model using learned filters from CDL-ACE. For CDL-ACE, we apply a practically feasible and convergent *Block Proximal Gradient method using Majorizer* (BPG-M)—which does not require difficult parameter tuning processes for convergence and acceleration—with a momentum coefficient formula and an adaptive restarting rule [5]. Numerical experiments show that, for strong additive white Gaussian (AWGN) noise, the image denoiser using learned filters by CDL-ACE significantly improves the quality of denoised images compared to Wiener filtering and total variation (TV) denoising methods. In particular, the learned filters by CDL-ACE improves image denoising compared to those trained by the conventional CDL, by resolving the model mismatch.

II. CDL-ACE: Model

The proposed (2D) CDL-ACE problem is given by the following joint optimization problem:

$$\begin{aligned} \min_{\{\mathbf{d}_k\},\{\mathbf{z}_{l,k}\},\{\boldsymbol{\rho}_l\}} \; & \sum_{l=1}^{L} \frac{1}{2} \left\| \mathbf{y}_l - \left(\mathbf{P}_B \sum_{k=1}^{K} \mathbf{d}_k \circledast \mathbf{z}_{l,k} \right) - \boldsymbol{\rho}_l \right\|_2^2 \\ & + \alpha \sum_{k=1}^{K} \left\| \mathbf{z}_{l,k} \right\|_1 + \frac{\beta}{2} \left\| \mathbf{C} \boldsymbol{\rho}_l \right\|_2^2 \\ \text{s.t.} \quad & \|\mathbf{d}_k\|_2^2 \le 1, \qquad k = 1, \ldots, K, \end{aligned} \tag{1}$$

where $\{\mathbf{d}_k \in \mathbb{R}^D : k = 1, \ldots, K\}$ is a set of convolutional filters to be learned, $\{\mathbf{y}_l \in \mathbb{R}^N : l = 1, \ldots, L\}$ is a set of training data, $\circledast$ denotes a circular convolution operator, $\{\mathbf{z}_{l,k} \in \mathbb{R}^N : l = 1, \ldots, L, k = 1, \ldots, K\}$ is a set of sparse codes, $\mathbf{P}_B \in \mathbb{R}^{N \times \hat{N}}$ is a projection matrix with $|B| = N$ and $N \le \hat{N}$ [3], [5], B is a list of distinct indices from the set $\{1, ..., \hat{N}\}$ that correspond to truncating the boundaries of the padded convolution $\sum_{k=1}^{K} \mathbf{d}_k \circledast \mathbf{z}_{l,k}$, $\{\boldsymbol{\rho}_l \in \mathbb{R}^N : l = 1, \ldots, L\}$ is a set of low-frequency component vectors, and $\mathbf{C} \in \mathbb{R}^{N' \times N}$ is a regularization transform for adaptive contrast enhancement of $\{\mathbf{y}_l\}$ (see below). Here, D is the filter size, K is the number of filters, N is the dimension of training data, $\hat{N}$ is the dimension after a convolution with padding, L is the number of training images, and N' is a dimension of signal in the transform $\mathbf{C}$ domain.[1] Note that $D \ll \hat{N}$ in general.

We reformulate problem (1) to a more convenient form. Considering that the minimizer with respect to $\boldsymbol{\rho}$ is:

$$\boldsymbol{\rho}_l(\{\mathbf{z}_{l,k}\}) = \left(\beta \mathbf{C}^T \mathbf{C} + \mathbf{I}_N \right)^{-1} \left(\mathbf{y}_l - \left(\mathbf{P}_B \sum_{k=1}^{K} \mathbf{d}_k \circledast \mathbf{z}_{l,k} \right) \right), \tag{2}$$

This work is supported in part by the Keck Foundation, UM-SJTU Collaborative Research Program, and NIH Grant U01 EB018753.

[1] In general, $\hat{(\cdot)}$ denotes a padded signal vector or its dimension.

978-1-5386-1565-2/17/$31.00 ©2017 IEEE

we rewrite problem (1) as

$$\begin{aligned}\min_{\{\mathbf{d}_k\},\{\mathbf{z}_{l,k}\}} &\sum_{l=1}^{L}\frac{1}{2}\left\|\tilde{\mathbf{y}}_l - \mathbf{R}\left(\mathbf{P}_B\sum_{k=1}^{K}\mathbf{d}_k \circledast \mathbf{z}_{l,k}\right)\right\|_2^2 + \alpha\sum_{k=1}^{K}\|\mathbf{z}_{l,k}\|_1 \\ \text{s.t.} \quad &\|\mathbf{d}_k\|_2^2 \leq 1, \qquad k=1,\ldots,K,\end{aligned} \tag{3}$$

where $\{\tilde{\mathbf{y}}_l := \mathbf{R}\mathbf{y}_l : l = 1,\ldots,L\}$ and

$$\mathbf{R} := \left(\beta\mathbf{C}^T\mathbf{C}\right)^{1/2}\left(\beta\mathbf{C}^T\mathbf{C}+\mathbf{I}\right)^{-1/2}. \tag{4}$$

Note that the reformulation (3) is valid with particular boundary conditions (e.g., periodic and reflective) for $\{\boldsymbol{\rho}_l\}$.[2] The filter $\mathbf{C}$ and its parameter β should be carefully designed so that the "filter" $\mathbf{R}$ in (4) can mimic the nonlinear contrast enhancement preprocessing widely used in CNN [1].

Using Parseval's relation [2], problem (3) is equivalent to the following joint optimization problem:

$$\begin{aligned}\min_{\{\mathbf{d}_k\},\{\hat{\mathbf{z}}_{l,k}\}} &\sum_{l=1}^{L}\frac{1}{2}\left\|\tilde{\mathbf{y}}_l - \mathbf{R}\mathbf{P}_B\sum_{k=1}^{K}\mathbf{\Phi}^H \mathrm{diag}(\tilde{\mathbf{d}}_k)\mathbf{\Phi}\hat{\mathbf{z}}_{l,k}\right\|_2^2 \\ &+\alpha\sum_{k=1}^{K}\|\hat{\mathbf{z}}_{l,k}\|_1, \quad \text{s.t.} \quad \|\mathbf{d}_k\|_2^2 \leq 1, \quad k=1,\ldots,K,\end{aligned} \tag{5}$$

where $\mathbf{\Phi}$ denotes the $\hat{N}$-point 2D unitary discrete Fourier transform, $\{\tilde{\mathbf{d}}_k = \sqrt{\hat{N}}\mathbf{\Phi}\mathbf{P}_S^T\mathbf{d}_k : k = 1,\ldots,K\}$ is a set of the frequency responses of the padded filters, $\mathbf{P}_S^T \in \mathbb{C}^{\hat{N}\times D}$ is zero-padding matrix, S is a list of indices that correspond to a small support of the filter with $|S| = D$, and $\{\hat{\mathbf{z}}_{l,k} \in \mathbb{C}^{\hat{N}} : l = 1,\ldots,L, k = 1,\ldots,K\}$ denotes sparse codes. Note that applying augmented Lagrangian methods (e.g., alternating direction method of multipliers) [2]–[5] to solve (5) requires one more auxiliary variable for each of $\{\mathbf{d}_k\}$ and $\{\hat{\mathbf{z}}_l\}$ updates. In other words, one faces even trickier parameter tuning processes. We apply BPG-M [5] which guarantees convergence and avoids parameter tuning (except the regularization parameters α, β).

III. CDL-ACE: CONVERGENT ALGORITHM

A. Convergent Fast BPG-M with Adaptive Restarting

1) Convergent Fast BPG-M: This section reviews the setup of *block multi-convex* problem and summarizes fast BPG-M [5]. Consider the optimization problem

$$\min_{\mathbf{x}\in\mathcal{X}} \; F(\mathbf{x}_1,\ldots,\mathbf{x}_B) := f(\mathbf{x}_1,\ldots,\mathbf{x}_B) + \sum_{b=1}^{B} r_b(\mathbf{x}_b) \tag{6}$$

where variable $\mathbf{x}$ is decomposed into B blocks $\mathbf{x}_1,\ldots,\mathbf{x}_B$ ($\{\mathbf{x}_b \in \mathbb{R}^{n_b}\}$), the set $\mathcal{X}$ of feasible points is assumed to be closed and *block multi-convex* subset of $\mathbb{R}^n$, f is assumed to be a differentiable and *block multi-convex* function, and r_b are extended-value convex functions for $b = 1,\ldots,B$.[3] We use $r_1,\ldots,r_B$ to enforce individual constraints of $\mathbf{x}_1,\ldots,\mathbf{x}_B$, when they are present. Importantly, r_b can include nonsmooth functions.

We are particularly interested in adopting the following quadratic *majorizer* (i.e., surrogate function) model of the composite function $\varrho(\mathbf{u}) = \varrho_1(\mathbf{u}) + \varrho_2(\mathbf{u})$ at a given point $\mathbf{v}$ to the block multi-convex problem (6):

$$\begin{aligned}\hat{\varrho}_M(\mathbf{u},\mathbf{v}) &= \psi_M(\mathbf{u};\mathbf{v}) + \varrho_2(\mathbf{u}), \\ \psi_M(\mathbf{u};\mathbf{v}) &= \varrho_1(\mathbf{v}) + \langle\nabla\varrho_1(\mathbf{v}), \mathbf{u}-\mathbf{v}\rangle + \frac{1}{2}\|\mathbf{u}-\mathbf{v}\|_{\mathbf{M}}^2\end{aligned} \tag{7}$$

where $\varrho_1(\mathbf{u})$ and $\varrho_2(\mathbf{u})$ are two convex functions defined on the convex set $\mathcal{U}$, $\varrho_1(\mathbf{u})$ is differentiable, and $\mathbf{M} = \mathbf{M}^T \succ 0$ is so-called *majorization matrix*.

Based on majorizers of the form (7), fast BPG-M [5] is given as follows. To solve (6), we minimize F cyclically over each block $\mathbf{x}_1,\ldots,\mathbf{x}_B$, while fixing the remaining blocks at their previously updated values. Let $\mathbf{x}_b^{(i+1)}$ be the value of $\mathbf{x}_b$ after its ith update, and

$$f_b^{(i)}(\mathbf{x}_b) := f(\mathbf{x}_1^{(i+1)},\ldots,\mathbf{x}_{b-1}^{(i+1)},\mathbf{x}_b,\mathbf{x}_{b+1}^{(i)},\ldots,\mathbf{x}_B^{(i)}), \tag{8}$$

for all b, i. The corresponding proximal mapping is given by

$$\begin{aligned}&\mathrm{Prox}_{r_b}\left(\acute{\mathbf{x}}_b^{(i)} - \left(\mathbf{M}_b^{(i)}\right)^{-1}\nabla f_b^{(i)}(\acute{\mathbf{x}}_b^{(i)}); \mathbf{M}_b^{(i)}\right) \\ &= \operatorname*{argmin}_{\mathbf{x}_b\in\mathcal{X}_b^{(i)}} \langle\nabla f_b^{(i)}(\acute{\mathbf{x}}_b^{(i)}), \mathbf{x}_b - \acute{\mathbf{x}}_b^{(i)}\rangle + \frac{1}{2}\left\|\mathbf{x}_b - \acute{\mathbf{x}}_b^{(i)}\right\|_{\mathbf{M}_b^{(i)}}^2 + r_b(\mathbf{x}_b),\end{aligned}$$

where $\nabla f_b^{(i)}(\acute{\mathbf{x}}_b^{(i)})$ is the block-partial gradient of f at $\acute{\mathbf{x}}_b^{(i)}$, $\mathcal{X}_b^{(i)} = \mathcal{X}_b(\mathbf{x}_1^{(i+1)},\ldots,\mathbf{x}_{b-1}^{(i+1)},\mathbf{x}_{b+1}^{(i)},\ldots,\mathbf{x}_B^{(i)})$, $\mathbf{M}_b^{(i)} \in \mathbb{R}^{n_b\times n_b}$ is a symmetric positive definite majorization matrix for the Hermitian $\nabla^2 f_b^{(i)}(\mathbf{x}_b) \succeq 0$, $\forall\mathbf{x}_b \in \mathcal{X}_b^{(i)}$, and the proximal operator is defined by $\mathrm{Prox}_r(\mathbf{y};\mathbf{M}) := \mathrm{argmin}_{\mathbf{x}}\,\frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_{\mathbf{M}}^2 + r(\mathbf{x})$. The matrix $\mathbf{W}_b^{(i)} \in \mathbb{R}^{n_b\times n_b}$, bounded by

$$0 \preceq \mathbf{W}_b^{(i)} \preceq \delta\left(\mathbf{M}_b^{(i)}\right)^{-1/2}\left(\mathbf{M}_b^{(i-1)}\right)^{1/2}, \quad \delta < 1, \; \forall b, i, \tag{9}$$

is an *extrapolation matrix* [5] that significantly accelerates convergence, in a similar manner to the extrapolation weight introduced in [11]. Similar to [12], we apply some momentum coefficient formulas $w^{(i)}$ to the extrapolation matrix updates $\mathbf{W}_b^{(i)}$ to accelerate BPG-M [5]:

$$w^{(i+1)} = \frac{\theta^{(i)}-1}{\theta^{(i+1)}}, \qquad \theta^{(i+1)} = \frac{1+\sqrt{1+4(\theta^{(i)})^2}}{2} \tag{10}$$

This choice guarantees fast convergence of fast proximal gradient (FPG) method in [13]. For diagonal majorization

[2] Throughout the paper, we apply the reflective boundary condition for (4), i.e., $\mathbf{C}^T\mathbf{C}$ and $\mathbf{R}$ in (4) are decomposed by cosine basis. We observed that the reflective boundary condition improves about 0.1 dB peak signal-to-noise ratio (PSNR) in image denoising applications.

[3] A set $\mathcal{X}$ is called *block multi-convex* if its projection to each block of variable is convex, i.e., for each b and any fixed $B-1$ blocks $\mathbf{x}_1,\ldots,\mathbf{x}_{b-1},\mathbf{x}_{b+1},\ldots,\mathbf{x}_B$, the set $\mathcal{X}_b(\mathbf{x}_1,\ldots,\mathbf{x}_{b-1},\mathbf{x}_{b+1},\ldots,\mathbf{x}_B) := \{\mathbf{x}_b : (\mathbf{x}_1,\ldots,\mathbf{x}_{b-1},\mathbf{x}_b,\mathbf{x}_{b+1},\ldots,\mathbf{x}_B) \in \mathcal{X}\}$ is convex. A function f is called *block multi-convex* if for each b, f is a convex function of x_b, when all the other blocks are fixed. Extended-value means $r_b(\mathbf{x}_b) = \infty$ if $\mathbf{x}_b \notin \mathrm{dom}(r_b)$, for $b = 1,\ldots,B$. In particular, r_b can be indicator functions of convex sets.

Algorithm 1 FBPG-M using a diagonal majorizer

```
Require: ${\mathbf{x}_b^{(1)} = \mathbf{x}_b^{(0)} : b = 1, \ldots, B}$, $\omega \in [-1, 0]$, $i = 1$
  while a stopping criterion is not satisfied do
    for $b = 1, \ldots, B$ do
      Calculate $\mathbf{M}_b^{(i)}$ for $\mathbf{f}_b^{(i)}(x_b)$ in (8)
      Calculate $\mathbf{W}_b^{(i)}$ by (11) with $\mathbf{M}_b^{(i)}, \mathbf{M}_b^{(i-1)}, w^{(i)}$
      $\acute{\mathbf{x}}_b^{(i)} = \mathbf{x}_b^{(i)} + \mathbf{W}_b^{(i)} \left( \mathbf{x}_b^{(i)} - \mathbf{x}_b^{(i-1)} \right)$
      $\mathbf{x}_b^{(i+1)} = \mathrm{Prox}_{r_b}\Big( \acute{\mathbf{x}}_b^{(i)} - \left( \mathbf{M}_b^{(i)} \right)^{-1} \nabla f_b^{(i)}(\acute{\mathbf{x}}_b^{(i)}); \mathbf{M}_b^{(i)} \Big)$
    end for
    Update $w^{(i+1)}$ by (10)
    $i = i + 1$
  end while
```

matrices $\mathbf{M}_b^{(i)}$, $\mathbf{M}_b^{(i-1)}$, the extrapolation matrix update is given by

$$\left(\mathbf{W}_b^{(i)} \right)_{j,j} = \delta \cdot \min \left\{ w^{(i)}, \left(\left(\mathbf{M}_b^{(i)} \right)^{-1} \mathbf{M}_b^{(i-1)} \right)_{j,j}^{1/2} \right\}, \tag{11}$$

where $\delta < 1$ appeared in (9), for $j = 1, \ldots, n_b$. We refer to BPG-M combined with the modified extrapolation matrix updates (11) using the momentum coeff. (10) as *Fast BPG-M* (FBPG-M). Algorithm 1 summarizes FBPG-M's updates.

Note that, under some mild conditions (e.g., continuity, lower-boundedness, and existence of critical points of F, etc [5, *Assumptions* 1–3]), any limit point of $\{\mathbf{x}^{(i)}\}$ generated from Algorithm 1 is a stationary point; see details of the convergence analysis in [5]. Applying (F)BPG-M to CDL(-ACE) provides the first convergence guarantee in CDL. In addition, (F)BPG-M methods gave lower objective values than the block coordinate descent scheme in [2] for CDL [5], [11].

2) Restarting Fast BPG-M: To further accelerate FBPG-M, we apply the adaptive momentum restarting scheme introduced in [14] (after the proximal mapping problem is solved in Algorithm 1). This technique restarts the algorithm when a restarting criterion is satisfied:

$$\acute{\mathbf{x}}_b^{(i)} = \mathbf{x}_b^{(i)}; \tag{12}$$

$$\mathbf{x}_b^{(i+1)} = \mathrm{Prox}_{r_b}\left(\acute{\mathbf{x}}_b^{(i)} - \left(\mathbf{M}_b^{(i)} \right)^{-1} \nabla f_b^{(i)}(\acute{\mathbf{x}}_b^{(i)}); \mathbf{M}_b^{(i)} \right). \tag{13}$$

We adopt a *gradient-mapping* criterion (referred to *reG*) [5]:

$$\cos\left(\Theta\left(\mathbf{M}_b^{(i)} \left(\acute{\mathbf{x}}_b^{(i)} - \mathbf{x}_b^{(i+1)} \right), \mathbf{x}_b^{(i+1)} - \mathbf{x}_b^{(i)} \right) \right) > \omega, \tag{14}$$

where the angle between two real vectors $\boldsymbol{\vartheta}$ and $\boldsymbol{\vartheta}'$ is given by $\Theta(\boldsymbol{\vartheta}, \boldsymbol{\vartheta}') := \langle \boldsymbol{\vartheta}, \boldsymbol{\vartheta}' \rangle / \left(\|\boldsymbol{\vartheta}\|_2 \|\boldsymbol{\vartheta}'\|_2 \right)$, and $\omega \in [-1, 0]$.

To solve the biconvex problem (5), we apply Algorithm 1 with adaptive restarting (12)–(13), promoting stable and fast convergence. The following sections present separable majorizers and introduce efficient proximal mapping methods.

B. Convolutional Dictionary (Filter) Update

1) Separable Majorizer Design: Using the current estimates of the $\{\hat{\mathbf{z}}_l : l = 1, \ldots, L\}$, the filter update problem for (5) can be rewritten by

$$\min_{\{\mathbf{d}_k\}} \frac{1}{2} \left\| \begin{bmatrix} \tilde{\mathbf{y}}_1 \\ \vdots \\ \tilde{\mathbf{y}}_L \end{bmatrix} - \boldsymbol{\Psi} \begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_K \end{bmatrix} \right\|_2^2, \text{ s.t. } \begin{array}{c} \|\mathbf{d}_k\|_2^2 \leq 1, \\ k = 1, \ldots, K \end{array}, \tag{15}$$

where

$$\boldsymbol{\Psi} := \left(\mathbf{I}_L \otimes \mathbf{R} \mathbf{P}_B \boldsymbol{\Phi}^H \right) \widetilde{\mathbf{Z}} \left(\mathbf{I}_K \otimes \sqrt{\hat{N}} \boldsymbol{\Phi} \mathbf{P}_S^T \right), \tag{16}$$

$$\widetilde{\mathbf{Z}} := \begin{bmatrix} \mathrm{diag}(\tilde{\mathbf{z}}_{1,1}) & \cdots & \mathrm{diag}(\tilde{\mathbf{z}}_{1,K}) \\ \vdots & \ddots & \vdots \\ \mathrm{diag}(\tilde{\mathbf{z}}_{L,1}) & \cdots & \mathrm{diag}(\tilde{\mathbf{z}}_{L,K}) \end{bmatrix} \in \mathbb{C}^{L\hat{N} \times K\hat{N}}.$$

We now design a block separable majorizer for the Hessian matrix $\boldsymbol{\Psi}^T \boldsymbol{\Psi} \in \mathbb{R}^{KD \times KD}$ of the cost function in (15). Observe that *1)* $\mathbf{R}^T \mathbf{R} \preceq \lambda_{\max} \mathbf{I}_N$ with $\lambda_{\max} = \max_{j=1,\ldots,N} \Lambda_{j,j}$, where $\mathbf{R}^T \mathbf{R} = \boldsymbol{\Xi}^T \boldsymbol{\Lambda} \boldsymbol{\Xi}$, $\boldsymbol{\Xi} \in \mathbb{R}^{N \times N}$ is a 2D discrete cosine transform, and $\boldsymbol{\Lambda}$ is a diagonal matrix; and *2)* $\boldsymbol{\Phi} \mathbf{P}_B^T \mathbf{P}_B \boldsymbol{\Phi}^H \preceq \mathbf{I}_{\hat{N}}$. Using these two bounds, $\boldsymbol{\Psi}^T \boldsymbol{\Psi}$ is bounded by

$$\boldsymbol{\Psi}^T \boldsymbol{\Psi} \preceq \lambda_{\max} \hat{N} \cdot \left(\mathbf{I}_K \otimes \mathbf{P}_S \right) \mathbf{Q}_\Psi^H \mathbf{Q}_\Psi \left(\mathbf{I}_K \otimes \mathbf{P}_S^T \right), \tag{17}$$

where $\mathbf{Q}_\Psi^H \mathbf{Q}_\Psi \in \mathbb{C}^{K\hat{N} \times K\hat{N}}$ is a block matrix with submatrices $\{[\mathbf{Q}_\Psi^H \mathbf{Q}_\Psi]_{k,k'} \in \mathbb{C}^{\hat{N} \times \hat{N}} : k, k' = 1, \ldots, K\}$:

$$[\mathbf{Q}_\Psi^H \mathbf{Q}_\Psi]_{k,k'} := \boldsymbol{\Phi}^H \sum_{l=1}^{L} \mathrm{diag}(\tilde{\mathbf{z}}_{l,k}^* \odot \tilde{\mathbf{z}}_{l,k'}) \boldsymbol{\Phi}. \tag{18}$$

Based on the bound (17), we design a diagonal majorization matrix for $\boldsymbol{\Psi}^T \boldsymbol{\Psi}$ using the following Lemma.

Lemma 3.1 (Block diagonal majorization matrix $\mathbf{M}_\Psi$)**.** *The following block diagonal matrix with diagonal blocks,* $\mathbf{M}_\Psi \in \mathbb{R}^{KD \times KD}$*, satisfies* $\mathbf{M}_\Psi \succeq \boldsymbol{\Psi}^T \boldsymbol{\Psi}$*:*

$$\mathbf{M}_\Psi = \hat{N} \lambda_{\max} \cdot \mathrm{diag}\left((\mathbf{I}_K \otimes \mathbf{P}_S) |\mathbf{Q}_\Psi^H \mathbf{Q}_\Psi| \left(\mathbf{I}_K \otimes \mathbf{P}_S^T \right) \mathbf{1}_{KD} \right),$$

where $\mathbf{Q}_\Psi^H \mathbf{Q}_\Psi$ *is defined in (18) and* $|\mathbf{A}|$ *denotes the matrix consisting of the absolute values of the elements of* $\mathbf{A}$*.*

2) Proximal Mapping: Because our majorization matrix in Lemma 3.1 is block diagonal, the proximal mapping problem (15) simplifies to separate problems for each filter:

$$\mathbf{d}_k^{(i+1)} = \underset{\mathbf{d}_k}{\mathrm{argmin}} \frac{1}{2} \left\| \mathbf{d}_k - \boldsymbol{\nu}_k^{(i)} \right\|_{\left[\mathbf{M}_\Psi^{(i)} \right]_{k,k}}^2, \text{ s.t. } \|\mathbf{d}_k\|_2^2 \leq 1, \tag{19}$$

where $\boldsymbol{\nu}^{(i)} = \acute{\mathbf{d}}^{(i)} - \left(\mathbf{M}_\Psi^{(i)} \right)^{-1} \left(\boldsymbol{\Psi}^{(i)} \right)^T \left(\boldsymbol{\Psi}^{(i)} \acute{\mathbf{d}}^{(i)} - \tilde{\mathbf{y}} \right)$, $\boldsymbol{\Psi}^{(i)}$ is defined in (16) with updated sparse codes $\{\hat{\mathbf{z}}_l^{(i)} : l = 1, \ldots, L\}$, $\mathbf{M}_\Psi^{(i)} \succeq (\boldsymbol{\Psi}^{(i)})^T \boldsymbol{\Psi}^{(i)}$ by Lemma 3.1, $\tilde{\mathbf{y}}$ is a concatenated vector with $\{\tilde{\mathbf{y}}_l\}$, and $\boldsymbol{\nu}^{(i)}$ is a concatenated vector with $\{\boldsymbol{\nu}_k^{(i)} \in \mathbb{R}^D : k = 1, \ldots, K\}$ ($\acute{\mathbf{d}}^{(i)}$ is constructed similar to $\boldsymbol{\nu}^{(i)}$). We apply accelerated Newton's method to efficiently solve (19) [5].

C. Sparse Code Update

1) Separable Majorizer Design: Given the current estimates of the filters $\{\tilde{\mathbf{d}}_k = \sqrt{\hat{N}} \boldsymbol{\Phi} \mathbf{P}_S^T \mathbf{d}_k : k = 1, \ldots, K\}$, the sparse code update problem for (5) is given by

$$\min_{\{\hat{\mathbf{z}}_l\}} \sum_{l=1}^{L} \frac{1}{2} \|\tilde{\mathbf{y}}_l - \boldsymbol{\Gamma} \hat{\mathbf{z}}_l\|_2^2 + \alpha \|\hat{\mathbf{z}}_l\|_1, \tag{20}$$

where $\hat{\mathbf{z}}_l = [\hat{\mathbf{z}}_{l,1}^T, \ldots, \hat{\mathbf{z}}_{l,K}^T]^T \in \mathbb{R}^{K\hat{N}}$ and

$$\boldsymbol{\Gamma} := \mathbf{R}\mathbf{P}_B \left[\boldsymbol{\Phi}^H \mathrm{diag}(\tilde{\mathbf{d}}_1) \boldsymbol{\Phi} \ \cdots \ \boldsymbol{\Phi}^H \mathrm{diag}(\tilde{\mathbf{d}}_K) \boldsymbol{\Phi} \right]. \tag{21}$$

We now seek a block separable majorizer for the Hessian matrix $\boldsymbol{\Gamma}^T\boldsymbol{\Gamma} \in \mathbb{C}^{K\hat{N} \times K\hat{N}}$ of the quadratic term in (20). Similar to (17), $\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}$ is bounded by $\boldsymbol{\Gamma}^T\boldsymbol{\Gamma} \preceq \lambda_{\max} \cdot \mathbf{Q}_\Gamma^H \mathbf{Q}_\Gamma$, where $\mathbf{Q}_\Gamma^H \mathbf{Q}_\Gamma \in \mathbb{C}^{K\hat{N} \times K\hat{N}}$ is a block matrix with submatrices $\{[\mathbf{Q}_\Gamma^H \mathbf{Q}_\Gamma]_{k,k'} \in \mathbb{C}^{\hat{N} \times \hat{N}} : k, k' = 1, \ldots, K\}$:

$$[\mathbf{Q}_\Gamma^H \mathbf{Q}_\Gamma]_{k,k'} = \boldsymbol{\Phi}^H \mathrm{diag}(\tilde{\mathbf{d}}_k^* \odot \tilde{\mathbf{d}}_{k'}) \boldsymbol{\Phi}. \tag{22}$$

We design a diagonal majorization matrix for $\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}$ as follows:

Lemma 3.2 (Block diagonal majorization matrix $\mathbf{M}_\Gamma$)**.** *The following block diagonal matrix with diagonal blocks,* $\mathbf{M}_\Gamma \in \mathbb{R}^{K\hat{N} \times K\hat{N}}$*, satisfies* $\mathbf{M}_\Gamma \succeq \boldsymbol{\Gamma}^T\boldsymbol{\Gamma}$*:*

$$\mathbf{M}_\Gamma = \mathrm{diag}\big(|\mathbf{Q}_\Gamma^H \mathbf{Q}_\Gamma| \mathbf{1}_{K\hat{N}} \big),$$

where $\mathbf{Q}_\Gamma^H \mathbf{Q}_\Gamma$ *is defined in (22).*

2) Proximal Mapping: The corresponding proximal mapping problem of (20) is separable for each training image:

$$\hat{\mathbf{z}}_l^{(i+1)} = \operatorname*{argmin}_{\hat{\mathbf{z}}_l} \frac{1}{2} \left\| \hat{\mathbf{z}}_l - \boldsymbol{\zeta}_l^{(i)} \right\|_{\mathbf{M}_\Gamma^{(i)}}^2 + \alpha \|\hat{\mathbf{z}}_l\|_1, \tag{23}$$

where $\boldsymbol{\zeta}_l^{(i)} = \acute{\mathbf{z}}_l^{(i)} - \left(\mathbf{M}_\Gamma^{(i)}\right)^{-1} \left(\boldsymbol{\Gamma}^{(i)}\right)^T \left(\boldsymbol{\Gamma}^{(i)} \acute{\mathbf{z}}_l^{(i)} - \tilde{\mathbf{y}}_l\right)$, $\boldsymbol{\Gamma}^{(i)}$ is defined in (21) with updated kernels $\{\mathbf{d}_k^{(i)} : k = 1, \ldots, K\}$, $\mathbf{M}_\Gamma^{(i)} \succeq (\boldsymbol{\Gamma}^{(i)})^T \boldsymbol{\Gamma}^{(i)}$ by Lemma 3.2, and $\boldsymbol{\zeta}_l^{(i)}$ is a concatenated vector with $\{\boldsymbol{\zeta}_{l,k}^{(i)} \in \mathbb{R}^{\hat{N}} : k = 1, \ldots, K\}$, for $l = 1, \ldots, L$. The solution to (23) is efficiently computed by soft-shrinkage [5], [15]. Note that one needs not to use $\boldsymbol{\Gamma}^{(i)}$ directly to compute $\boldsymbol{\zeta}_l^{(i)}$. If the filter size D is smaller than $\log \hat{N}$, it is more efficient to use (circular) convolutions.

IV. Application of Learned Filters by CDL-ACE to Image Denoising

To denoise a measured image $\mathbf{b} \in \mathbb{R}^n$ corrupted by AWGN ($\sim \mathcal{N}(0, \sigma^2)$), we use the filters $\{\mathbf{d}_k^\star : k = 1, \ldots, K\}$ learned via (3) while solving the following optimization problem:

$$\begin{aligned} \{\{\mathbf{a}_k^\star\}, \boldsymbol{\rho}^\star\} = \operatorname*{argmin}_{\{\mathbf{a}_k\}, \boldsymbol{\rho}} \frac{1}{2} \left\| \mathbf{b} - \left(\mathbf{P}_B \sum_{k=1}^{K} \mathbf{d}_k^\star \circledast \mathbf{a}_k \right) - \boldsymbol{\rho} \right\|_2^2 \\ + \alpha' \sum_{k=1}^{K} \|\mathbf{a}_k\|_1 + \beta' \|\mathbf{C}\boldsymbol{\rho}\|_2^2, \end{aligned} \tag{24}$$

and synthesize the denoised image by $\mathbf{x}^\star = \mathbf{P}_B \sum_{k=1}^K \mathbf{d}_k^\star \circledast \mathbf{a}_k^\star + \boldsymbol{\rho}^\star$, where $\boldsymbol{\rho}^\star$ is constructed similar to (2). Using the reformulation techniques in (3) and (5), we rewrite (24) as the following convex problem:

$$\hat{\mathbf{a}}^\star = \operatorname*{argmin}_{\hat{\mathbf{a}} \in \mathbb{R}^{K\hat{n}}} \frac{1}{2} \left\| \tilde{\mathbf{b}} - \mathbf{A}^\star \hat{\mathbf{a}} \right\|_2^2 + \alpha' \|\hat{\mathbf{a}}\|_1, \tag{25}$$

where $\tilde{\mathbf{b}} = \mathbf{R}\mathbf{b}$, $\mathbf{A}^\star = \mathbf{R}\mathbf{P}_B \widehat{\mathbf{D}}^\star$, $\widehat{\mathbf{D}}^\star \hat{\mathbf{a}} = \sum_{k=1}^K \mathbf{d}_k^\star \circledast \mathbf{a}_k$, and $\hat{\mathbf{a}} = [\hat{\mathbf{a}}_1^T, \ldots, \hat{\mathbf{a}}_K^T]^T$. We solve (25) through FPG using a diagonal majorizer and adaptive restarting; see, for example, [15, Fig. 2]. Note that (24) is the first image denoising model based on CDL for non-contrast-enhanced noisy images.

V. Numerical Results and Discussion

A. Experimental Setup

Using FBPG-M (Algorithm 1) with adaptive restarting rule (14)[4] [5, reG-FBPG-M], we trained 100 and 200 filters with the conventional CDL (i.e., (1) with $\{\boldsymbol{\rho}_l = \mathbf{0}\}$ and $\beta = 0$ [3], [5]) and the proposed CDL-ACE (1) for the city datasets [1], [2]. For the conventional CDL, we trained filters with preprocessed (intensity rescaling to $[0, 1]$ and local contrast enhancement [1]) and non-preprocessed datasets (note that both the datasets contain zero-mean training images). For the preprocessed and non-preprocessed datasets, we set $\alpha = 1$ and $\alpha = 0.4$, respectively. For CDL-ACE, we trained filters with non-preprocessed datasets (note that even substracting the mean is not required). We used the second-order finite difference for $\mathbf{C}$ in (1) with $\beta = 25$. The chosen regularization parameters provide a good balance between sparsity and data fitting terms. We terminated the iterations if either of the following stopping criteria are met before reaching the maximum number of iterations 3,000 [3]:

$$\begin{aligned} & F(\mathbf{d}^{(\text{iter}+1)}, \hat{\mathbf{z}}^{(\text{iter})}), F(\mathbf{d}^{(\text{iter}+1)}, \hat{\mathbf{z}}^{(\text{iter}+1)}) \geq F(\mathbf{d}^{(\text{iter})}, \hat{\mathbf{z}}^{(\text{iter})}) \\ \text{or} \quad & \frac{\left\| \mathbf{d}^{(\text{iter}+1)} - \mathbf{d}^{(\text{iter})} \right\|_2}{\left\| \mathbf{d}^{(\text{iter}+1)} \right\|_2}, \frac{\left\| \hat{\mathbf{z}}^{(\text{iter}+1)} - \hat{\mathbf{z}}^{(\text{iter})} \right\|_2}{\left\| \hat{\mathbf{z}}^{(\text{iter}+1)} \right\|_2} < \mathrm{tol}, \end{aligned} \tag{26}$$

where $(\cdot)^{(\text{iter}+1)}$ denotes the updated variable $(\cdot)$ after its iter$^{\text{th}}$ update, and $\mathbf{d}$ and $\hat{\mathbf{z}}$ are concatenated vectors from $\{\mathbf{d}_k\}$ and $\{\hat{\mathbf{z}}_{l,k}\}$, respectively. The tolerance value, tol, was set to 10^{-5}.

For image denoising applications, we corrupted a test image with strong AWGN, i.e., SNR $= 5, 10$ dB. We denoised the noisy image through the following methods: *1)* adaptive Wiener filtering with 3×3 window size; *2)* TV with MFISTA using its regularization parameter 0.8σ and maximum number of iterations 200 [16]; *3)* the proposed image denoising model (24) with 200 learned filters by the conventional CDL and preprocessed training data, $\alpha' = 2.5\sigma$, the first-order finite difference for $\mathbf{C}$ in (24) [10], and $\beta' = 10\sigma$; and *4)* (24) with 200 learned filters by CDL-ACE (1), $\alpha' = \alpha \cdot 5.5\sigma$, and $\beta' = \beta \cdot 5.5\sigma$ (i.e., scaled regularization parameters used in training). For (24), the majorization matrix $\mathbf{M} \succeq (\mathbf{A}^\star)^T \mathbf{A}^\star$ is designed by [5, Lem. 4.12]; and the stopping criteria is set similar to (26) (with $\mathrm{tol} = 2.5 \times 10^{-2}, 10^{-3}$ for $\mathrm{SNR}_{\mathrm{dB}} = 5, 10$, respectively) before reaching the maximum number of iterations 100. We tuned all the parameters in the introduced image denoising methods to give the best PSNR values.

B. Convolutional Dictionary Learning with Adaptive Contrast Enhancement

Fig. 2(b) shows that CDL-ACE successfully learns filters with "Gabor-like" shapes that are not fully learned by the

[4] We selected the parameter ω in (14) as $\cos(95°)$ [5].

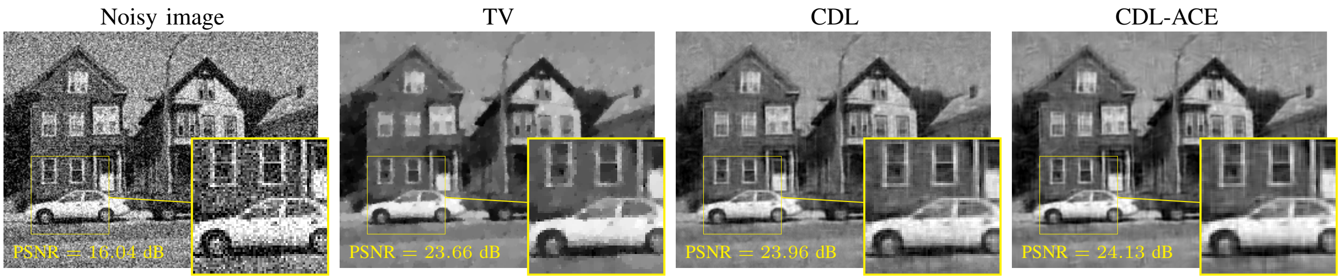
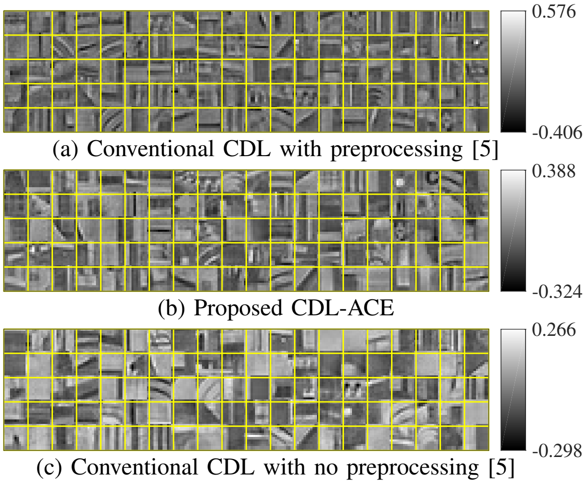


Fig. 1. Comparison of denoised images from different image denoising models (image is corrupted by AWGN with SNR = 10 dB). The proposed image denoising model (24) using learned filters from CDL outperforms Wiener filtering and TV denoising. The learned filters by CDL-ACE further improves (24).



(a) Conventional CDL with preprocessing [5]

(b) Proposed CDL-ACE

(c) Conventional CDL with no preprocessing [5]

Fig. 2. Examples of learned filters with different CDL models (100 filters).

TABLE I

PSNR values (dB) of different image denoising models with different SNRs in noisy images

SNR	Noisy	Wiener	TV	CDL	CDL-ACE
5 dB	11.04	17.81	21.28	21.43	**21.68**
10 dB	16.04	22.02	23.66	23.96	**24.13**

conventional CDL using non-preprocessed training datasets, e.g., Fig. 2(c). More carefully designed transform $\mathbf{C}$ in (1) is expected to better capture high-frequency components of training datasets by more closely mimicking the nonlinear contrast enhancement through $\mathbf{R}$ in (4).

C. Image Denoising with Learned Convolutional Dictionaries

The proposed image denoising model (24) using the learned filters through CDL-ACE significantly improves PSNR over Wiener filtering and TV image denoising; see Table I and Fig. 1. In particular, the proposed CDL-ACE successfully resolves the model mismatch between the training and testing models—compare CDL and CDL-ACE in Table I and Fig. 1—thereby improving the quality of denoised images in particular for stronger noise (i.e., $\mathrm{SNR}_{\mathrm{dB}} = 5$). Replacing ℓ^1 with a nonconvex penality can avoid the bias of soft-shrinkage and suppress the artifacts in uniform regions (e.g., sky).

VI. Conclusion

The proposed CDL-ACE is the first CDL model that bridges the gap between CDL and its application to inverse problems, by successfully modeling preprocessing and learning Gabor-like shapes in filters. For strong AWGN, learned filters by CDL-ACE further improves the proposed image denoising model compared to those trained by the conventional CDL, i.e., it improves PSNR by approximately 0.2 dB. Future work will explore the effectiveness of learned filters from CDL-ACE to image reconstruction problems with extremely undersampled or noisy measurements.

References

[1] M. D. Zeiler *et al.*, "Deconvolutional networks," in *Proc.* 2010 *IEEE CVPR*, San Francisco, CA, Jun. 2010, pp. 2528–2535.

[2] H. Bristow *et al.*, "Fast convolutional sparse coding," in *Proc.* 2013 *IEEE CVPR*, Portland, OR, Jun. 2013, pp. 391–398.

[3] F. Heide *et al.*, "Fast and flexible convolutional sparse coding," in *Proc.* 2015 *IEEE CVPR*, Boston, MA, Jun. 2015, pp. 5135–5143.

[4] B. Wohlberg, "Efficient algorithms for convolutional sparse representations," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 301–315, Jan. 2016.

[5] I. Y. Chun and J. A. Fessler, "Convolutional dictionary learning: Acceleration and convergence," submitted to *IEEE Trans. Image Process.*, Nov. 2016.

[6] A. M. Bruckstein *et al.*, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, Feb. 2009.

[7] J. Mairal *et al.*, "Sparse modeling for image and vision processing," *Found. & Trends in Comput. Graph. Vis.*, vol. 8, no. 2–3, pp. 85–283, Dec. 2014.

[8] K. Kavukcuoglu *et al.*, "Learning convolutional feature hierarchies for visual recognition," in *Proc. NIPS* 2010*, Advances in Neural Information Processing Systems* 23, 2010, pp. 1090–1098.

[9] V. Papyan *et al.*, "Convolutional neural networks analyzed via convolutional sparse coding," *arXiv preprint cs.ML:1607.08194*, 2016.

[10] A. Serrano *et al.*, "Convolutional sparse coding for high dynamic range imaging," vol. 35, no. 2, pp. 153–163, May 2016.

[11] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM J. Imaging Sci.*, vol. 6, no. 3, pp. 1758–1789, Sep. 2013.

[12] ——, "A fast patch-dictionary method for whole image recovery," *Inverse Probl. Imag.*, vol. 10, no. 2, pp. 563–583, May 2016.

[13] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, Mar. 2009.

[14] P. Giselsson and S. Boyd, "Monotonicity and restart in fast gradient methods," in *Proc.* 53^{rd} *IEEE CDC*, Los Angeles, CA, Dec. 2014, pp. 5058–5063.

[15] M. J. Muckley *et al.*, "Fast parallel MR image reconstruction via B1-based, adaptive restart, iterative soft thresholding algorithms (BARISTA)," *IEEE Trans. Med. Imag.*, vol. 34, no. 2, pp. 578–588, Feb. 2015.

[16] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, Nov. 2009.