

# Optimized Color Filter Arrays for Sparse Representation-Based Demosaicking

Jia Li, Chenyan Bai, Zhouchen Lin, *Senior Member, IEEE*, and Jian Yu

**Abstract**—Demosaicking is the problem of reconstructing a color image from the raw image captured by a digital color camera that covers its only imaging sensor with a color filter array (CFA). Sparse representation-based demosaicking has been shown to produce superior reconstruction quality. However, almost all existing algorithms in this category use the CFAs, which are not specifically optimized for the algorithms. In this paper, we consider optimally designing CFAs for sparse representation-based demosaicking, where the dictionary is well-chosen. The fact that CFAs correspond to the projection matrices used in compressed sensing inspires us to optimize CFAs via minimizing the mutual coherence. This is more challenging than that for traditional projection matrices because CFAs have physical realizability constraints. However, most of the existing methods for minimizing the mutual coherence require that the projection matrices should be unconstrained, making them inapplicable for designing CFAs. We consider directly minimizing the mutual coherence with the CFA’s physical realizability constraints as a generalized fractional programming problem, which needs to find sufficiently accurate solutions to a sequence of nonconvex nonsmooth minimization problems. We adapt the redistributed proximal bundle method to address this issue. Experiments on benchmark images testify to the superiority of the proposed method. In particular, we show that a simple sparse representation-based demosaicking algorithm with our specifically optimized CFA can outperform LSSC [1]. To the best of our knowledge, it is the first sparse representation-based demosaicking algorithm that beats LSSC in terms of CPSNR.

**Index Terms**—Color filter array (CFA), demosaicking, sparse representation, mutual coherence, generalized fractional programming, redistributed proximal bundle method, linearized alternating direction method with parallel splitting and adaptive penalty (LADMPSAP).

Manuscript received November 9, 2016; revised February 23, 2017; accepted March 4, 2017. Date of publication March 7, 2017; date of current version March 27, 2017. The work of J. Li and J. Yu was supported by the National Natural Science Foundation of China (NSFC) under Grant 61632004 and Grant 61370129. The work of Z. Lin was supported in part by the 973 Program of China under Grant 2015CB352502 and in part by the NSFC under Grant 61625301 and Grant 61231002. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jie Liang.

J. Li and J. Yu are with the Beijing Key Laboratory of Traffic Data Analysis and Mining, School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China (e-mail: jiali.gm@gmail.com; jianyu@bjtu.edu.cn).

C. Bai is with the College of Information Engineering and Beijing Advanced Innovation Center for Imaging Technology, Capital Normal University, Beijing 100048, China (e-mail: cybai@cnu.edu.cn).

Z. Lin is with the Key Laboratory of Machine Perception (Ministry of Education), School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China, and also with the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zlin@pku.edu.cn).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2017.2679440

## I. INTRODUCTION

AT EACH pixel color images usually consist of three color components, e.g., red (R), green (G), and blue (B), or cyan (C), magenta (M), and yellow (Y). Consequently, a digital color camera needs three separate sensors, one for each color component. However, it is expensive to use three sensors and it is difficult to precisely align all sensors. So most digital color cameras utilize only a single sensor and place a color filter array (CFA) (see Fig. 1) in front of the sensor, which results in the captured images having only one color component at each pixel. Such images are called raw images. One has to use a demosaicking algorithm to reconstruct color images from raw images. Accordingly, the quality of the finally produced color images depends on both the CFA and the demosaicking algorithm.

On the one hand, developing a tailored demosaicking algorithm for a given CFA to well exploit its characteristics has been studied extensively [2]–[4]. For example, directional interpolation based demosaicking [5]–[8] interpolates along multiple directions to efficiently utilize both the interchannel and the intrachannel correlations. Frequency selection based demosaicking [9]–[11] takes advantage of the spectral characteristics of raw images. Sparse representation based demosaicking [1], [12], [13] considers demosaicking as an inverse problem and exploits sparsity prior by decomposing each image patch into a sparse representation over a dictionary. However, these demosaicking algorithms usually use the predefined CFAs, which are designed empirically with different preferences. For instance, the most popular Bayer CFA [14] was designed to mimic human visual system’s greatest response to green light (Fig. 1(a)). Sony company [15] replaced a green component of the Bayer CFA pattern with an emerald (E) component, which is claimed to be closer to human color perception. Lukac and Plataniotis [16] developed the Lukac CFA using only R, G, and B, where green pixels are in pairs and vertically adjacent in each pair (Fig. 1(b)). As the panchromatic pixel does not block the visible light, Compton and Hamilton [17] included panchromatic pixels in the CFA to improve light sensitivity (Fig. 1(c)). The recently designed Fujifilm X-Trans CFA [18] was inspired by the claim that the irregular arrangement of particles in traditional film is advantageous in reducing Moiré artifacts.

On the other hand, different types of demosaicking algorithms provide different compromises between reconstruction efficiency and accuracy [24], [25]. So they have different application scenarios. To enhance the performance of a given

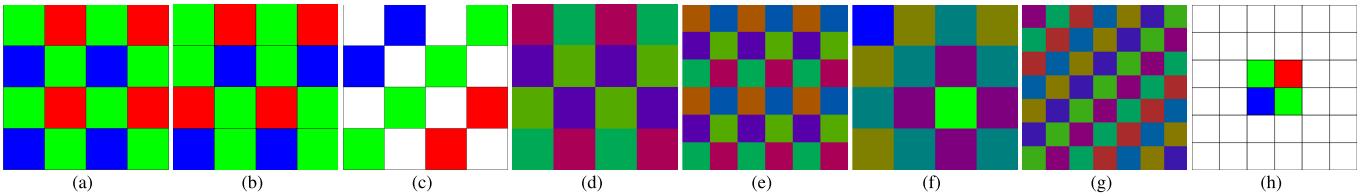


Fig. 1. Eight representative CFA patterns. They are the (a) Bayer CFA [14], (b) Lukac CFA [16], (c) Compton CFA [17], (d) Hirakawa CFA [19], (e) Condat CFA [20], (f) Hao CFA [21], (g) Diag7 CFA [22], and (h) Chakrabarti CFA [23], respectively. The sum across color channels of every CFA is normalized to be an all-one matrix. **Images in this paper are best viewed on screen!**

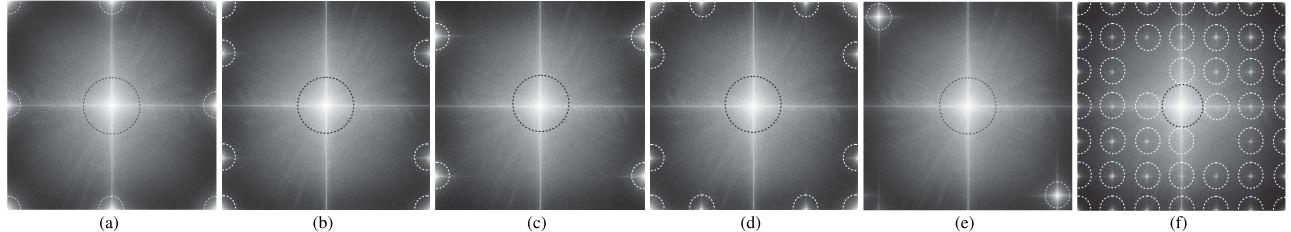


Fig. 2. The spectra of raw images by six CFAs. They are the average spectra of all raw images from the Kodak dataset [32] and their corresponding CFAs are the (a) Bayer CFA [14], (b) Hirakawa CFA [19], (c) Condat CFA [20], (d) Hao CFA [21], (e) Diag7 CFA [22], and (f) Chakrabarti CFA [23], respectively. In each image, the luminance component is located in the center and on the horizontal and the vertical axes, which is denoted by the black dashed circle, while the chrominance components are denoted by the white dashed circles.

type of demosaicking algorithms, one can design specific CFAs for it, where the demosaicking process is fully considered in the CFA design. For example, frequency selection based demosaicking is very attractive for its linearity, which gives a good trade-off between computational complexity and reconstruction quality. Accordingly, the CFA design in the frequency domain [19]–[22], [26] optimizes CFAs specifically for it. Spectral characteristic analysis [9], [19], [21] shows that a raw image by a periodic CFA can be interpreted in the frequency domain as the sum of a nonsubsampled component at the baseband and multiple subsampled replicas of components at the high frequency bands, where the nonsubsampled component is called luminance component (denoted by the black dashed circle in Fig. 2) and the subsampled components are called chrominance components (denoted by the white dashed circles in Fig. 2). The different locations of frequency components lead to frequency selection based demosaicking. Frequency selection based demosaicking first uses appropriate bandpass filters to estimate all frequency components. Then according to the transformation from luminance/chrominance basis to RGB basis, it recovers the color image in RGB format from the frequency components. So the CFA design in the frequency domain needs to minimize spectral overlap to obtain more accurate estimations of all frequency components. Also, it needs to maximize the numerical stability of color transformation. These two design criteria have been discussed in detail [22]. We show the spectra of raw images by the Bayer CFA and four optimized CFAs for frequency selection based demosaicking in Fig. 2(a) and 2(b)–2(e), respectively. We can see that the luminance and chrominance components of the Bayer CFA overlap on the horizontal and the vertical axes, while those of the four optimized CFAs have no spectral overlap.

Recently, the Chakrabarti CFA [23] was proposed specifically for its associated inpainting and colorization reconstruction algorithm. In order to be compatible with

inpainting and colorization techniques, the  $6 \times 6$  Chakrabarti CFA pattern embedded the  $2 \times 2$  Bayer CFA pattern in the center and set the remaining pixels be panchromatic ones (see Fig. 1(h)). Accordingly, the reconstruction algorithm first recovered the missing panchromatic values at the Bayer CFA pattern, which is an inpainting task. Then it estimated color information from the Bayer CFA pattern and propagated the color information to panchromatic pixels, which is a scribble based colorization problem. The Chakrabarti CFA demosaicked by its associated reconstruction algorithm has been shown to reduce noise and aliasing artifacts in low-light conditions. However, from the average spectrum of raw images by the Chakrabarti CFA shown in Fig. 2(f), we can see that there is severe spectral overlap. So the Chakrabarti CFA is not a favorable one for frequency selection based demosaicking. Conversely, the optimized CFAs for frequency selection based demosaicking (Fig. 1(d)–1(g)) are not preferred by the reconstruction algorithm associated with the Chakrabarti CFA either. So in order to enhance the performance of a type of demosaicking algorithms, we need to optimize the CFAs it uses.

Sparse representation based demosaicking has been shown to produce superior reconstruction quality [1], [12], [13]. However, almost all existing algorithms in this category concentrate only on dictionary design and directly use the predefined CFAs which are not specifically optimized for the demosaicking algorithms. For instance, Mairal *et al.* [12] extended the grayscale K-SVD denoising algorithm [27] to demosaic raw images by the Bayer CFA. They learned the dictionary on the vectorized color image patches. To reduce false color artifacts, they also proposed a new inner product used in the orthogonal matching pursuit (OMP) [28]. A simultaneous sparse representation based demosaicking algorithm was presented in [1], which encourages similar patches to have similar sparse representations. It also online learned an adaptive dictionary for every test image to further improve

demosaicking performance. Instead of learning a dictionary, Moghadam *et al.* [13] manually constructed the dictionary to reduce both the interchannel and the intrachannel correlations of color images. It is claimed to generate sparser representations and hence improves the performance of sparse representation based demosaicking. They also showed that their demosaicking algorithm with the Hirakawa CFA performs better than that with the Bayer CFA. However, like all existing sparse representation based demosaicking algorithms, their demosaicking algorithm also used the predefined CFAs and did not specifically optimize the CFAs they used. Note that the sparse representation based demosaicking discussed in the paper is based on conventional color cameras, where the only component of every pixel of a raw image is a combination of the RGB components of the corresponding color image at the same pixel. For those based on single-pixel color cameras [29], where each sequential measurement is a different combination of the RGB components of the color image at all pixels, we refer the readers to [30] and [31].

The newly emerged compressed sensing (CS) [33], [34] is a technique for joint signal sampling and compression. The CS theory shows that a signal which has a sparse representation over an appropriate overcomplete dictionary can be recovered from fewer linear projections than required by the traditional sampling theorem. The linear projections of a signal are the inner products between it and a set of projection vectors, arranged as the rows of a projection matrix. The product of a projection matrix and a dictionary is called the effective dictionary. The CS theory also shows that for a given number of linear projections and an appropriate dictionary, designing the projection matrix via minimizing the mutual coherence [35] of the effective dictionary could substantially improve recovery accuracy. Since sparse representation based demosaicking is a specific CS problem, based on the above theory, in this paper we consider optimizing CFAs specifically for it via minimizing the mutual coherence. This is more challenging than optimizing traditional CS projection matrices, which are usually assumed to be unconstrained. Most of the existing methods for optimizing CS projection matrices [36]–[40] work with the column normalized effective dictionaries. However, CFAs have constraints on their physical realizability, which conflict with the column normalization of effective dictionaries (see subsection II-B). As a result, these methods for optimizing CS projection matrices cannot be directly used to design CFAs.

The contributions of this paper are:

- Based on the theory of CS, we propose a new CFA design method for sparse representation based demosaicking, where the dictionary is well-chosen and fixed. To the best of our knowledge, it is the first CFA design method specifically for sparse representation based demosaicking, which is also theoretically grounded.
- We develop a new method to directly minimize the mutual coherence with the constraints of CFA's physical realizability, which is more challenging than that for unconstrained CS projection matrices.
- We demonstrate that a simple sparse representation based demosaicking algorithm with our specifically optimized

TABLE I  
SUMMARY OF THE MAIN NOTATIONS USED IN THIS PAPER

Notation	Definition
$\mathbb{R}, \mathbb{R}_+$	Set of real and nonnegative real numbers, respectively
$\mathcal{C}^2$	Set of twice differentiable functions
$\partial f(\mathbf{x})$	Set of subgradients of $f$ at $\mathbf{x}$ (Subdifferential of $f$ at $\mathbf{x}$ )
$\mathbf{1}$	All-one vector or matrix
$\mathbf{0}$	All-zero vector or matrix
$\mathbf{I}$	Identity matrix
$\mathbf{x}(i)$	$i$ -th element of vector $\mathbf{x}$
$\text{Diag}(\mathbf{x})$	Diagonal matrix whose $i$ -th diagonal element is $\mathbf{x}(i)$
$\langle \cdot, \cdot \rangle$	inner product between two matrices or vectors
$\mathbf{X}(i, j, k)$	$(i, j, k)$ -th element of third-order tensor $\mathbf{X}$
$\mathbf{X}(:, i:j)$	$i$ -th through $j$ -th columns of matrix $\mathbf{X}$
$\text{diag}(\mathbf{X})$	Vector whose $i$ -th element is $\mathbf{X}(i, i)$
$\text{vec}(\mathbf{X})$	Vector formed by stacking all the columns of matrix $\mathbf{X}$
$\mathbf{X}^T$	Transpose of matrix $\mathbf{X}$
$\ \mathbf{x}\ _0$	$\ell_0$ -norm (number of nonzero elements) of vector $\mathbf{x}$
$\ \mathbf{x}\ _2$	$\ell_2$ -norm $(\sqrt{\sum_i \mathbf{x}(i)^2})$ of vector $\mathbf{x}$
$\ \mathbf{X}\ _F$	Frobenius norm $(\sqrt{\sum_i \sum_j \mathbf{X}(i, j)^2})$ of matrix $\mathbf{X}$
$\ \mathbf{X}\ _\infty$	$\ell_\infty$ -norm $(\max_i \{\sum_j  \mathbf{X}(i, j) \})$ of matrix $\mathbf{X}$

Note that  $\ell_0$ -norm is not a true norm because it does not satisfy the positive scalability:  $\|a\mathbf{x}\|_0 = |a| \|\mathbf{x}\|_0$  for any scalar  $a$ . The term "norm" here is for convenience.

CFA can outperform LSSC [1]. To the best of our knowledge, we present the first sparse representation based demosaicking algorithm that beats LSSC in terms of CPSNR. Our demosaicking algorithm is also much more efficient.

The rest of the paper is organized as follows. In Section II, we introduce our model to design CFAs for sparse representation based demosaicking. Then we describe the solving process of our CFA design model in Section III. In Section IV, we conduct experiments to test our design method. Finally, we conclude the paper in Section V.

## II. OUR CFA DESIGN MODEL

In this section, we first introduce sparse representation based demosaicking, as we will design CFAs for it. Then we present our CFA design model, which is specifically for the sparse representation based demosaicking with a given dictionary.

We focus only on periodic CFAs defined on the square lattice, where the minimum periodic array is called the CFA pattern. For extensive reviews of nonperiodic CFAs, we refer to [41]. As color images are usually stored and processed in RGB format, we use the RGB color model to represent colors, i.e., every color is decomposed into a mixture of R, G, and B. Then designing a CFA pattern is to optimize the three mixing coefficients of every pixel of the CFA pattern. We use upper case boldface letters for matrices, lower case boldface letters for vectors, and lower case letters for scalars, respectively, where vectors are all column ones. Color image patches are rearranged as column vectors in the order of R, G, and B. The main notations used throughout the paper are summarized in Table I.

### A. Sparse Representation Based Demosaicking

In the noiseless case, the model of color imaging with a CFA is as follows:

$$\mathbf{y} = \mathbf{P}\mathbf{x} = (\mathbf{P}_R, \mathbf{P}_G, \mathbf{P}_B)(\mathbf{x}_R^T, \mathbf{x}_G^T, \mathbf{x}_B^T)^T, \quad (1)$$

where  $\mathbf{y} \in \mathbb{R}^{m \times 1}$  is the raw image patch,  $\mathbf{x} \in \mathbb{R}^{3m \times 1}$  is the color image patch, with  $\mathbf{x}_R$ ,  $\mathbf{x}_G$ , and  $\mathbf{x}_B$  being its red, green, and blue channels, respectively,  $\mathbf{P} \in \mathbb{R}^{m \times 3m}$  is the mosaicking matrix, and  $\mathbf{P}_R$ ,  $\mathbf{P}_G$ , and  $\mathbf{P}_B$  are diagonal matrices whose diagonal elements are specified by the red, green, and blue channels of the CFA, respectively. Formally, let the CFA be  $\mathbf{C} \in \mathbb{R}^{\sqrt{m} \times \sqrt{m} \times 3}$ , then  $\mathbf{P}_R = \text{Diag}(\text{vec}(\mathbf{C}(:, :, 1)))$ ,  $\mathbf{P}_G = \text{Diag}(\text{vec}(\mathbf{C}(:, :, 2)))$ , and  $\mathbf{P}_B = \text{Diag}(\text{vec}(\mathbf{C}(:, :, 3)))$ .

On the other hand, natural image statistics shows that a natural image patch can be sparsely represented by an appropriate overcomplete dictionary [42], which can be described as:

$$\mathbf{x} = \mathbf{D}\boldsymbol{\alpha}, \quad (2)$$

where  $\mathbf{D} = (\mathbf{d}_1, \dots, \mathbf{d}_n) \in \mathbb{R}^{3m \times n}$  is the dictionary, which is overcomplete (i.e.,  $n > 3m$ ) and column normalized to Euclidean unit length (i.e.,  $\|\mathbf{d}_i\|_2 = 1, i \in \{1, \dots, n\}$ ), and  $\boldsymbol{\alpha} \in \mathbb{R}^{n \times 1}$  is the representation coefficient and satisfies  $\|\boldsymbol{\alpha}\|_0 \ll n$ . Then according to the CS theory [33], [34], the color image patch  $\mathbf{x}$  can be recovered from the raw image patch  $\mathbf{y}$  in (1) by using the sparse representation in (2), i.e.,  $\mathbf{y} = \mathbf{P}\mathbf{D}\boldsymbol{\alpha}$  with the sparest  $\boldsymbol{\alpha}$ . More formally, for each raw image patch  $\mathbf{y}$ , sparse representation based demosaicking requires solving the following problem [1], [12], [13]:

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \|\boldsymbol{\alpha}\|_0, \quad \text{s.t. } \mathbf{y} = \mathbf{P}\mathbf{D}\boldsymbol{\alpha}. \quad (3)$$

It can be solved efficiently by using OMP [28] and the color image patch  $\mathbf{x}$  can be reconstructed as  $\hat{\mathbf{x}} = \mathbf{D}\boldsymbol{\alpha}^*$ . To demosaick a large raw image, sparse representation based demosaicking first demosaicks all the overlapping  $\sqrt{m} \times \sqrt{m}$  patches of the raw image. Then it fuses the overlapping demosaicked image patches to get the demosaicked image.

### B. Proposed Formulation

The recent work on CS demonstrates that using a well designed projection matrix rather than a random one, can improve the performance of signal recovery [36], [40]. For the sparse representation based demosaicking in (3), the mosaicking matrix  $\mathbf{P}$  is a specific CS projection matrix. Based on the CS theory, the mutual coherence [35] of  $\mathbf{PD}$  should be minimized. This enhances the uniqueness of the sparse solution to problem (3) and hence improves demosaicking performance. The mutual coherence of a matrix is defined as follows:

*Definition 1: Given a matrix  $\mathbf{A} = (\mathbf{a}_1, \dots, \mathbf{a}_n) \in \mathbb{R}^{m \times n}$ , its mutual coherence  $\mu(\mathbf{A})$  is defined as the largest correlation (in absolute value) between different columns of  $\mathbf{A}$ . Formally,*

$$\mu(\mathbf{A}) = \max_{1 \leq i < j \leq n} \frac{|\langle \mathbf{a}_i, \mathbf{a}_j \rangle|}{\|\mathbf{a}_i\|_2 \|\mathbf{a}_j\|_2}. \quad (4)$$

CS commonly assumes that the dictionary  $\mathbf{D}$  is fixed while the projection matrix  $\mathbf{P}$  is unconstrained. We denote  $\mathbf{M} = \mathbf{PD}$  as the effective dictionary and  $\tilde{\mathbf{M}}$  as the normalization of  $\mathbf{M}$  with all Euclidean unit columns. Then the largest absolute off-diagonal element of the Gram matrix  $\tilde{\mathbf{M}}^T \tilde{\mathbf{M}}$  will be  $\mu(\mathbf{M})$ . Accordingly, most of the existing methods [36]–[40] use the Gram matrix  $\tilde{\mathbf{M}}^T \tilde{\mathbf{M}}$  to optimize the unconstrained projection matrix  $\mathbf{P}$ . However, the mosaicking matrix  $\mathbf{P}$  is derived from a

physically realizable CFA. So it has the following constraints. First,  $\mathbf{P}$  should be real, nonnegative, and formed by three diagonal matrices. This requires that  $\mathbf{P} \geq \mathbf{0}^{m \times 3m}$  and  $\mathbf{P} \circ \mathbf{Q} = \mathbf{0}^{m \times 3m}$ , where  $\mathbf{Q} = \mathbf{1}_{m \times 3m} - (\mathbf{I}^{m \times m}, \mathbf{I}^{m \times m}, \mathbf{I}^{m \times m})$ , and  $\geq$  and  $\circ$  stand for the componentwise greater than or equal to and product, respectively. Second, the sum across the color channels of a CFA should be an all-one matrix [19]–[22]. So we have  $\mathbf{P}\mathbf{1}^{3m \times 1} = \mathbf{1}^{m \times 1}$ . The two constraints together ensure that  $\mathbf{P}$  is a valid mosaicking matrix. Then there are only  $2m$  unknowns to finally determine the mosaicking matrix  $\mathbf{P}$ . However, that all columns of  $\mathbf{PD}$  are Euclidean unit will generate  $n$  ( $n > 3m$ ) equations, which are more than the unknowns. This is an overdetermined problem and may not have a solution. Thus we cannot use the Gram matrix to find the optimal mosaicking matrix, making most of the existing methods for minimizing the mutual coherence [36]–[40] inapplicable.

We further assume that the CFA pattern is smaller than an image patch, i.e.,  $r \leq \sqrt{m}$  and  $c \leq \sqrt{m}$ , where  $r \times c$  and  $\sqrt{m} \times \sqrt{m}$  are the sizes of the CFA pattern and image patch, respectively. So we need to consider the periodicity of the CFA covering a  $\sqrt{m} \times \sqrt{m}$  image patch. Without loss of generality, we assume that  $\sqrt{m}$  is divisible by both  $r$  and  $c$ . Then we have  $\mathbf{1}^T \mathbf{PR} = \mathbf{0}^{1 \times 3(m-rc)}$ , where  $\mathbf{R} = \mathbf{T}(\mathbf{I}(:, 1 : 3(m-rc)) - \mathbf{I}(:, 3rc+1 : 3m)) \in \mathbb{R}^{3m \times 3(m-rc)}$ ,  $\mathbf{1} \in \mathbb{R}^{m \times 1}$ ,  $\mathbf{I} \in \mathbb{R}^{3m \times 3m}$ , and  $\mathbf{T} \in \mathbb{R}^{3m \times 3m}$  is a permutation matrix. Right multiplying the row vector  $\mathbf{1}^T \mathbf{P}$  by  $\mathbf{T}$  will permute its columns to obtain a new row vector whose elements are rearranged CFA-pattern-wise.

According to the above discussion, we formulate our CFA design model as:

$$\begin{aligned} \min_{\mathbf{P}} \quad & \max_{1 \leq i < j \leq n} \left\{ \frac{|\langle \mathbf{P}\mathbf{d}_i, \mathbf{P}\mathbf{d}_j \rangle|}{\|\mathbf{P}\mathbf{d}_i\|_2 \|\mathbf{P}\mathbf{d}_j\|_2} \right\} \\ \text{s.t. } \mathbf{P} \geq \mathbf{0}, \mathbf{P} \circ \mathbf{Q} = \mathbf{0}, \mathbf{P}\mathbf{1} = \mathbf{1}, \mathbf{1}^T \mathbf{PR} = \mathbf{0}. \end{aligned} \quad (5)$$

It is a generalized fractional programming problem [43].

### III. SOLVING OUR CFA DESIGN MODEL

In this section, we first briefly introduce generalized fractional programming. Then we detail the solution process of our CFA design model in (5).

#### A. Generalized Fractional Programming

The generalized fractional programming considers the following problem:

$$\lambda^* = \min_{\mathbf{x} \in \mathbb{S}} \max_{1 \leq i \leq q} \left\{ \frac{\varphi_i(\mathbf{x})}{\psi_i(\mathbf{x})} \right\}, \quad (6)$$

where  $\lambda^*$  is the optimal value of the problem,  $\mathbb{S}$  is the nonempty feasible set,  $\varphi_i$  and  $\psi_i$  are continuous functions on  $\mathbb{S}$ , and  $\psi_i$  is positive on  $\mathbb{S}$ . Crouzeix *et al.* [43], [44] proposed two Dinkelbach-type algorithms that consists of solving sequential problems in the form:

$$\gamma^{k+1} = \min_{\mathbf{x} \in \mathbb{S}} f^k(\mathbf{x}), \quad (7)$$

where  $f^k(\mathbf{x}) = \max_{1 \leq i \leq q} \{(\varphi_i(\mathbf{x}) - \lambda^k \psi_i(\mathbf{x})) / \omega_i^k\}$ ,  $\omega_i^k > 0$  is the normalization parameter, the optimal solution is denoted

as  $\mathbf{x}^{k+1}$ , and  $\lambda^k$  is the optimal objective function value of problem (6) at the previous iteration, i.e.,

$$\lambda^k = \max_{1 \leq i \leq q} \left\{ \frac{\varphi_i(\mathbf{x}^k)}{\psi_i(\mathbf{x}^k)} \right\}. \quad (8)$$

It has been shown that the sequence  $\{\lambda^k\}$  generated by (7) and (8) converges linearly to  $\lambda^*$  in (6) when  $\omega_i^k = 1$  [43], which can be improved to become superlinear when  $\omega_i^k = \psi_i(\mathbf{x}^k)$  [44]. Moreover, if  $\mathbb{S}$  is compact,  $\lambda^*$  in (6) is finite and every limit point of the sequence  $\{\mathbf{x}^k\}$  generated by either of the two algorithms is an optimal solution of problem (6).

However, the convergence of the above two algorithms requires finding an optimal solution to problem (7), which may not be easy, e.g.,  $f^k(\mathbf{x})$  in (7) is nonconvex and nonsmooth. To ease this requirement, some inexact Dinkelbach-type algorithms have been proposed, where only a sufficiently accurate solution of problem (7) is needed. For example, Birbil *et al.* [45] assumed that  $\mathbb{S}$  is convex and approximated  $f^k(\mathbf{x})$  in (7) from above, which produces a decreasing sequence  $\{\lambda^k\}$  by (8). Strodiot *et al.* [46] considered the case that problem (7) is convex, i.e.,  $f^k(\mathbf{x})$  and  $\mathbb{S}$  are both convex. Then they performed a serious step of the proximal bundle method [47] to construct a proper piecewise linear lower approximation of  $f^k(\mathbf{x})$ . They proved that the sequences  $\{\lambda^k\}$  and  $\{\mathbf{x}^k\}$  generated by these approximation problems converge to  $\lambda^*$  and an optimal solution of problem (6), respectively, where the convexity of  $f^k(\mathbf{x})$  is necessary to prove convergence. However, their algorithm cannot be directly applied to a nonconvex  $f^k(\mathbf{x})$ . This is because the proximal bundle method cannot ensure to obtain an accurate enough solution of problem (7) when  $f^k(\mathbf{x})$  is nonconvex. The proximal bundle method can generate a piecewise linear lower approximation to a convex  $f^k(\mathbf{x})$  with any desired accuracy, which is defined as the maximum of tangent lines. The lower approximation is measured by the linearization errors, which are the differences between  $f^k(\mathbf{x})$  and all tangent lines at the current prox-center. If  $f^k(\mathbf{x})$  is convex, the tangent lines are below  $f^k(\mathbf{x})$  and hence the linearization errors are always nonnegative. However, the linearization errors can be negative when  $f^k(\mathbf{x})$  is nonconvex, which implies that the tangent lines may cut off a region containing a minimum.

We focus on the situation that  $\mathbb{S}$  is compact and  $f^k(\mathbf{x})$  in (7) is nonconvex and nonsmooth, which is much harder. So we only consider developing an inexact Dinkelbach-type algorithm that can guarantee to generate a decreasing sequence  $\{\lambda^k\}$  by (8). Fortunately, this is tractable and can be guided by Theorem 1.

**Theorem 1:** Let the sequence  $\{\mathbf{x}^k\}$  be the feasible approximation solutions to problem (7). If  $f^k(\mathbf{x}^{k+1}) \leq 0$  is always met, the corresponding sequence  $\{\lambda^k\}$  generated by (8) is decreasing and convergent.

*Proof:* From the definition in (8), we have that there exists  $i' \in \{1, \dots, q\}$  such that

$$\lambda^{k+1} = \max_{1 \leq i \leq q} \left\{ \frac{\varphi_i(\mathbf{x}^{k+1})}{\psi_i(\mathbf{x}^{k+1})} \right\} = \frac{\varphi_{i'}(\mathbf{x}^{k+1})}{\psi_{i'}(\mathbf{x}^{k+1})}. \quad (9)$$

Since  $f^k(\mathbf{x})$  in (7) is the maximum of its  $q$  component functions, it is greater than or equal to any of its component

functions for all  $\mathbf{x} \in \mathbb{S}$ . In particular, the following inequality holds at  $\mathbf{x}^{k+1}$  for the given  $i' \in \{1, \dots, q\}$ :

$$\begin{aligned} f^k(\mathbf{x}^{k+1}) &\geq (\varphi_{i'}(\mathbf{x}^{k+1}) - \lambda^k \psi_{i'}(\mathbf{x}^{k+1})) / \omega_{i'}^k \\ &= (\lambda^{k+1} - \lambda^k) \psi_{i'}(\mathbf{x}^{k+1}) / \omega_{i'}^k. \end{aligned} \quad (10)$$

As  $\psi_i(\mathbf{x})$  and  $\omega_i^k$  are both positive on  $\mathbb{S}$ , if  $f^k(\mathbf{x}^{k+1}) \leq 0$  is true,  $\lambda^{k+1} \leq \lambda^k$  will hold and hence the sequence  $\{\lambda^k\}$  is decreasing. Since  $\mathbb{S}$  is compact,  $\lambda^*$  in (6) will be finite and is a lower bound of  $\{\lambda^k\}$ . Thus  $\{\lambda^k\}$  is convergent and the theorem is proved. ■

By combining (7) and (8), it is easy to see that  $f^k(\mathbf{x}^k) = 0$ . Then according to Theorem 1, we only need to obtain a feasible approximation solution to problem (7) that is better than  $\mathbf{x}^k$ . Moreover, if we further assume that  $\psi_i(\mathbf{x})/\omega_i^k$  has an upper bound, applying the Squeeze Theorem (also known as the Sandwich Theorem) to (10) we can obtain that  $f^k(\mathbf{x}^{k+1})$  converges to 0, which can be used as the stop criterion. Note that  $\omega_i^k = 1$  and  $\omega_i^k = \psi_i(\mathbf{x}^k)$  are the commonly used normalization parameters. We use  $\omega_i^k = 1$  in the paper. Consequently, if  $\psi_i(\mathbf{x})$  is upper bounded,  $f^k(\mathbf{x}^{k+1})$  will converge to 0.

### B. Solving Problem (5)

Based on the above analysis, we develop an inexact Dinkelbach-type algorithm to solve problem (5), which can guarantee that the objective function value in (5) is decreasing. We first denote the feasible set of problem (5) by  $\Omega = \{\mathbf{P} \in \mathbb{R}^{m \times 3m} | \mathbf{P} \geq \mathbf{0}, \mathbf{P} \circ \mathbf{Q} = \mathbf{0}, \mathbf{P}\mathbf{1} = \mathbf{1}, \mathbf{1}^T \mathbf{P}\mathbf{R} = \mathbf{0}\}$ , which is compact and convex. Then at the  $k$ -th iteration, we need to solve the following problem with sufficient accuracy:

$$\min_{\mathbf{P} \in \Omega} f^k(\mathbf{P}), \quad (11)$$

where the sufficiently accurate solution is denoted as  $\mathbf{P}^{k+1}$ , the corresponding function value  $f^k(\mathbf{P}^{k+1})$  is denoted as  $\gamma^{k+1}$ ,

$$f^k(\mathbf{P}) = \max_{1 \leq i < j \leq n} \left\{ |\langle \mathbf{P}\mathbf{d}_i, \mathbf{P}\mathbf{d}_j \rangle| - \lambda^k \|\mathbf{P}\mathbf{d}_i\|_2 \|\mathbf{P}\mathbf{d}_j\|_2 \right\}, \quad (12)$$

and  $\lambda^k$  is the objective function value of problem (5) at  $\mathbf{P}^k$ , i.e.,

$$\lambda^k = \max_{1 \leq i < j \leq n} \left\{ |\langle \mathbf{P}^k \mathbf{d}_i, \mathbf{P}^k \mathbf{d}_j \rangle| / (\|\mathbf{P}^k \mathbf{d}_i\|_2 \|\mathbf{P}^k \mathbf{d}_j\|_2) \right\}. \quad (13)$$

Since  $\|\mathbf{P}\|_\infty = 1$  on  $\Omega$  and  $\|\mathbf{d}_i\|_2 = 1$ , we have that  $\|\mathbf{P}\mathbf{d}_i\|_2 \leq \|\mathbf{P}\|_2 \|\mathbf{d}_i\|_2 \leq \sqrt{m} \|\mathbf{P}\|_\infty = \sqrt{m}$  for all  $i \in \{1, \dots, n\}$ . Accordingly, for every  $1 \leq i < j \leq n$ ,  $\|\mathbf{P}\mathbf{d}_i\|_2 \|\mathbf{P}\mathbf{d}_j\|_2$  is bounded from above by  $m$ . So the sequence  $\{\gamma^k\}$  generated by solving problem (11) with sufficient accuracy converges to 0.

The whole process for solving problem (5) is summarized in Algorithm 1.

Since  $f^k(\mathbf{P})$  in (11) is nonconvex and nonsmooth, finding a sufficiently accurate solution of problem (11) is not easy. However,  $\langle \mathbf{P}\mathbf{d}_i, \mathbf{P}\mathbf{d}_j \rangle$  and  $\|\mathbf{P}\mathbf{d}_i\|_2 \|\mathbf{P}\mathbf{d}_j\|_2$  are  $C^2$  functions w.r.t.  $\mathbf{P}$  and hence lower- $C^2$  functions.<sup>1</sup> Functions defined

<sup>1</sup>A function  $f$  is lower- $C^2$  on an open set  $\mathcal{O}$  if  $f$  is finite on  $\mathcal{O}$  and for any  $\mathbf{x} \in \mathcal{O}$  there exists a threshold  $\kappa_{\min} > 0$  such that  $f(\mathbf{y}) + \frac{\kappa}{2} \|\mathbf{y} - \mathbf{x}\|_2^2$  is convex w.r.t.  $\mathbf{y}$  on an open neighbourhood of  $\mathbf{x}$  for all  $\kappa \geq \kappa_{\min}$ .

**Algorithm 1** Algorithm for Solving Problem (5)**Input:**  $\mathbf{D} = (\mathbf{d}_1, \dots, \mathbf{d}_n)$ ,  $m, r \times c$ .1: Initialization:  $\mathbf{P}^0 \in \Omega$ ,  $\gamma^0 = -1$ ,  $k = 0$ .2: **while**  $\gamma^k \neq 0$  **do**3:   compute  $\lambda^k$  by (13).4:   compute  $\mathbf{P}^{k+1}$  and  $\gamma^{k+1}$  by solving problem (11) with sufficient accuracy.5:    $k \leftarrow k + 1$ .6: **end while****Output:**  $\mathbf{P}^k$ .

by sums, absolute values, and maximums of lower- $C^2$  functions are lower- $C^2$  [48]. Thus  $f^k(\mathbf{P})$  in (11) is globally lower- $C^2$ . Accordingly, we adapt the redistributed proximal bundle method [49] to get an accurate enough solution of problem (11). The convergence of the redistributed proximal bundle method for solving the unconstrained nonconvex nonsmooth minimization with a lower- $C^2$  objective function has been well established. We detail the solution process of inexactly solving problem (11) in the following subsections.

**C. Local Convexification**

We first briefly introduce the local convexification strategy, which is the main idea of the redistributed proximal bundle method [49]. The redistributed proximal bundle method considers the following unconstrained minimization problem:

$$\min_{\mathbf{y}} f(\mathbf{y}), \quad (14)$$

where  $f(\mathbf{y})$  is a lower- $C^2$  function and may be nonconvex nonsmooth. As a proximal bundle method, it first augments  $f(\mathbf{y})$  with a stabilization term:

$$\min_{\mathbf{y}} f(\mathbf{y}) + \frac{\kappa}{2} \|\mathbf{y} - \mathbf{x}\|_2^2, \quad (15)$$

where  $\mathbf{x}$  and  $\kappa > 0$  are called the prox-center and prox-parameter, respectively. In contrast to other nonconvex bundle methods, it then splits the prox-parameter  $\kappa$  into two nonnegative terms  $\eta$  and  $\nu$  satisfying  $\kappa = \eta + \nu$ . Thus problem (15) could be equivalently rewritten as:

$$\min_{\mathbf{y}} \tilde{f}(\mathbf{y}) + \frac{\nu}{2} \|\mathbf{y} - \mathbf{x}\|_2^2, \quad (16)$$

where  $\tilde{f}(\mathbf{y}) = f(\mathbf{y}) + \frac{\eta}{2} \|\mathbf{y} - \mathbf{x}\|_2^2$  is assumed to be locally convex at  $\mathbf{x}$ . So a piecewise linear lower approximation to  $\tilde{f}(\mathbf{y})$  at  $\mathbf{x}$  can be constructed by using the cutting plane method [50]. Consequently, problem (16) is approximated from below by an unconstrained convex quadratic programming problem, which has efficient solving methods [51]. Thus solving problem (14) consists of solving a sequence of unconstrained convex quadratic programming problems. This strategy is called local convexification. Accordingly,  $\eta$  and  $\nu$  are referred as the convexification parameter and model prox-parameter, respectively, and have to be suitably modified during the iteration.

**D. Solving Problem (11) With Sufficient Accuracy**

We use the local convexification strategy to solve problem (11) with sufficient accuracy, which is a linearly constrained nonconvex nonsmooth minimization problem. Let  $t$  be the iteration counter and it depends on  $k$ . For notational simplicity, we omit the superscript  $k$  below.

At the  $t$ -th iteration, the current prox-center is denoted by  $\hat{\mathbf{P}}^t$  and the bundle information is formed by  $\{\mathbf{P}^l, f(\mathbf{P}^l), \mathbf{G}^l \in \partial f(\mathbf{P}^l)\}_{l=0}^t$ . Then by the local convexification strategy,  $\tilde{f}^t(\mathbf{P}) = f(\mathbf{P}) + \frac{\eta^t}{2} \|\mathbf{P} - \hat{\mathbf{P}}^t\|_F^2$  is bounded from below at  $\hat{\mathbf{P}}^t$  by the model function  $\varphi^t(\mathbf{P})$ , which is a piecewise linear function. Formally,

$$\begin{aligned} \varphi^t(\mathbf{P}) &= \max_{0 \leq l \leq t} \left\{ \tilde{f}^t(\mathbf{P}^l) + \langle \mathbf{G}^l + \eta^t(\mathbf{P}^l - \hat{\mathbf{P}}^t), \mathbf{P} - \mathbf{P}^l \rangle \right\} \\ &= f(\hat{\mathbf{P}}^t) + \max_{0 \leq l \leq t} \left\{ \langle \mathbf{G}^l + \eta^t(\mathbf{P}^l - \hat{\mathbf{P}}^t), \mathbf{P} - \hat{\mathbf{P}}^t \rangle - \tilde{e}_l^t \right\}, \end{aligned} \quad (17)$$

where  $\tilde{e}_l^t = e_l^t + \frac{\eta^t}{2} \|\mathbf{P}^l - \hat{\mathbf{P}}^t\|_F^2$  is called the augmented linearization error, and  $e_l^t = f(\hat{\mathbf{P}}^t) - (f(\mathbf{P}^l) + \langle \mathbf{G}^l, \hat{\mathbf{P}}^t - \mathbf{P}^l \rangle)$  is called the linearization error. Accordingly, the linearly constrained convex quadratic programming problem at the  $t$ -th iteration is:

$$\min_{\mathbf{P} \in \Omega} \varphi^t(\mathbf{P}) + \frac{\nu}{2} \|\mathbf{P} - \hat{\mathbf{P}}^t\|_F^2. \quad (18)$$

In order to improve the readability, we provide the solution process for (18) in the Appendix.

We compute the subgradient  $\mathbf{G}^l \in \partial f(\mathbf{P}^l)$  as follows. We compute a subgradient of the  $i,j$ -th component function of  $f(\mathbf{P})$  in (12) by

$$\begin{aligned} h_{ij}(\mathbf{P}) &= \text{sign}(\langle \mathbf{P} \mathbf{d}_i, \mathbf{P} \mathbf{d}_j \rangle) \mathbf{P} \left( \mathbf{d}_i \mathbf{d}_j^T + \mathbf{d}_j \mathbf{d}_i^T \right) \\ &\quad - \lambda \left( \frac{\|\mathbf{P} \mathbf{d}_i\|_2 \mathbf{P} \mathbf{d}_i^T}{\|\mathbf{P} \mathbf{d}_i\|_2} + \frac{\|\mathbf{P} \mathbf{d}_j\|_2 \mathbf{P} \mathbf{d}_j^T}{\|\mathbf{P} \mathbf{d}_j\|_2} \right), \end{aligned} \quad (19)$$

where  $\text{sign}(\cdot)$  is the sign function which extracts the sign of a real number. Suppose that the  $i,j$ -th component function of  $f(\mathbf{P})$  achieves the maximum at  $\mathbf{P}^l$ . Then we obtain the subgradient  $\mathbf{G}^l$  of  $f(\mathbf{P})$  at  $\mathbf{P}^l$  by

$$\mathbf{G}^l = h_{i'j'}(\mathbf{P}^l). \quad (20)$$

In order to guarantee that  $\varphi^t(\mathbf{P})$  is a lower approximation to  $\tilde{f}^t(\mathbf{P})$  at  $\hat{\mathbf{P}}^t$ , the augmented linearization error  $\tilde{e}_l^t$  in (17) should be nonnegative, i.e.,  $\tilde{e}_l^t \geq 0$ ,  $l \in \{0, \dots, t\}$ . Thus before the update of model function  $\varphi^t(\mathbf{P})$ , we compute  $\eta^t$  by

$$\eta^t = \max \left( \max_{0 \leq l \leq t, \mathbf{P}^l \neq \hat{\mathbf{P}}^t} \frac{-2e_l^t}{\|\mathbf{P}^l - \hat{\mathbf{P}}^t\|_F^2}, 0 \right), \quad (21)$$

which is the minimal value that keeps  $\tilde{e}_l^t$  nonnegative for all  $l \in \{0, \dots, t\}$ .

After the candidate prox-center  $\mathbf{P}^{t+1}$  is computed by (18), we check whether it provides sufficient decrease of  $f(\mathbf{P})$  as compared to  $\hat{\mathbf{P}}^t$ , which is measured by a fixed fraction of the predicted decrease:

$$\delta^t = f(\hat{\mathbf{P}}^t) - \left( \varphi^t(\mathbf{P}^{t+1}) + \frac{\nu}{2} \|\mathbf{P}^{t+1} - \hat{\mathbf{P}}^t\|_F^2 \right). \quad (22)$$

**Algorithm 2** Algorithm for Inexactly Solving Problem (11)**Input:**  $\hat{\mathbf{P}}^0$ ,  $\nu > 0$ .

- 1: Initialization:  $\mathbf{P}^0 = \hat{\mathbf{P}}^0$ ,  $\rho = 0.9$ ,  $\varepsilon = 10^{-8}$ ,  $\delta^0 = \varepsilon + 1$ ,  $t = 0$ .
- 2: compute  $f(\mathbf{P}^0)$ .
- 3: **while**  $\delta^t \geq \varepsilon$  **do**
- 4:   compute subgradient  $\mathbf{G}^t \in \partial f(\mathbf{P}^t)$  by (20).
- 5:   compute  $\eta^t$  by (21).
- 6:   update model function  $\varphi^t(\mathbf{P})$  by (17).
- 7:   compute  $\mathbf{P}^{t+1}$  by solving problem (18).
- 8:   compute  $f(\mathbf{P}^{t+1})$ .
- 9:   **if**  $f(\mathbf{P}^t) < f(\mathbf{P}^{t+1})$  **then**
- 10:     **return**  $(\hat{\mathbf{P}}^t, \text{true})$ .
- 11:   **end if**
- 12:   compute predicted decrease  $\delta^t$  by (22).
- 13:   **if**  $f(\hat{\mathbf{P}}^t) - f(\mathbf{P}^{t+1}) \geq \rho \delta^t$  **then**
- 14:     serious step:  $\hat{\mathbf{P}}^{t+1} \leftarrow \mathbf{P}^{t+1}$ ,  $f(\hat{\mathbf{P}}^{t+1}) \leftarrow f(\mathbf{P}^{t+1})$ .
- 15:   **else**
- 16:     null step:  $\hat{\mathbf{P}}^{t+1} \leftarrow \hat{\mathbf{P}}^t$ ,  $f(\hat{\mathbf{P}}^{t+1}) \leftarrow f(\hat{\mathbf{P}}^t)$ .
- 17:   **end if**
- 18:    $t \leftarrow t + 1$ .
- 19: **end while**
- 20: **return**  $(\hat{\mathbf{P}}^t, \text{false})$ .

**Output:**  $(\hat{\mathbf{P}}^*, \text{isrestart}) = \text{RPBM}(\hat{\mathbf{P}}^0, \nu)$ .**Algorithm 3** Algorithm for Inexactly Solving Problem (11) With Restarts**Input:**  $\mathbf{P}^k$ ,  $\nu = 5 \times 10^{-3}$ ,  $\xi = 2$ .

- 1: Initialization:  $\mathbf{P} = \mathbf{P}^k$ ,  $\text{isrestart} = \text{true}$ .
- 2: **while**  $\text{isrestart}$  **do**
- 3:    $(\mathbf{P}, \text{isrestart}) = \text{RPBM}(\mathbf{P}, \nu)$  by calling Algorithm 2.
- 4:    $\nu \leftarrow \nu \xi$ .
- 5: **end while**
- 6:  $\mathbf{P}^{k+1} = \mathbf{P}$ ,  $\gamma^{k+1} = f^k(\mathbf{P}^{k+1})$ .

**Output:**  $\mathbf{P}^{k+1}$  and  $\gamma^{k+1}$ .

Note that by combining (17), (18), and (21), we have that  $\varphi^t(\mathbf{P}^{t+1}) + \frac{\nu}{2} \|\mathbf{P}^{t+1} - \hat{\mathbf{P}}^t\|_F^2 \leq \varphi^t(\hat{\mathbf{P}}^t) \leq f(\hat{\mathbf{P}}^t)$ . So  $\delta^t$  is nonnegative. If the decrease is sufficient, we perform a serious step, where  $\mathbf{P}^{t+1}$  will be the next prox-center. Otherwise, we perform a null step and the next prox-center is still  $\hat{\mathbf{P}}^t$ .

Since  $f(\mathbf{P})$  is nonconvex,  $\mathbf{P}^{t+1}$  may increase  $f(\mathbf{P})$  as compared to  $\mathbf{P}^t$ , i.e.,  $f(\mathbf{P}^t) < f(\mathbf{P}^{t+1})$ . As in [49], when this happens, we restart the algorithm by initializing  $\mathbf{P}$  with the current prox-center  $\hat{\mathbf{P}}^t$  and setting  $\rho$  as  $\rho \xi$  ( $\xi > 1$ ). As  $f(\mathbf{P})$  is globally lower- $C^2$ , it has been proved that there is only a finite number of such restarts.

We summarize the whole inexact solution process of problem (11) in Algorithm 3.

## IV. EXPERIMENTS

In this section, we conduct experiments on benchmark test images to verify the effectiveness of our CFA design method. We use the Plain Sparse Representation based Demosaicking

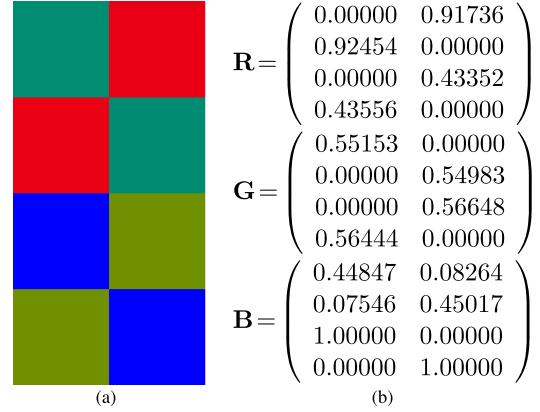


Fig. 3. Our specifically optimized CFA pattern for the publicly available dictionary  $\mathbf{D}$  [53]. (a) is our newly designed  $4 \times 2$  CFA pattern and (b) is its color values.

algorithm to test our designed CFAs, which we call PSRD. For a given dictionary, it demosaicks every  $\sqrt{m} \times \sqrt{m}$  raw image patch via solving (3) by OMP [28] and then simply averages the values of overlapping pixels of the demosaicked image patches to produce the demosaicked image, where nearby patches overlap  $\sqrt{m} - 1$  pixels. We use the `mexOMP` function in the SPAMS toolbox [52] as the implementation of OMP.

We design a new CFA for a well-chosen dictionary using the proposed method. Then we first compare it with the predefined CFAs, all demosaicked by PSRD using the chosen dictionary. We next compare our new CFA demosaicked by PSRD using the chosen dictionary with the state-of-the-art demosaicking. The two comparisons together can thoroughly testify to the effectiveness of our design method.

## A. Experimental Settings

1) *Dictionary and Dataset:* As we design CFAs for the sparse representation based demosaicking with a given dictionary, we need to choose the dictionary. In our experiments, we directly use the publicly available dictionary  $\mathbf{D} \in \mathbb{R}^{192 \times 256}$  [53]. The main reason is that state-of-the-art sparse representation based demosaicking algorithms [1], [12], [41] use the dictionary. When comparing with these algorithms using an identical dictionary, we can directly conclude that the performance gain of our approach results from the optimized CFA rather than the dictionary used. The dictionary is learned on  $2 \times 10^7$  color image patches with a size of  $8 \times 8$  pixels. These patches are randomly taken from the 9,963 natural images of the PASCAL VOC'07 dataset [54], which have no overlap with demosaicking benchmark images. Thus we have  $m = 8 \times 8$  and  $n = 256$  in (3). Accordingly, we also exclude an 8-pixel border (one patch width) to eliminate the boundary effect.

Demosaicking performances are usually evaluated on both the Kodak [32] and IMAX [55] datasets. The images in the Kodak dataset have weak channel correlations, while those in the IMAX dataset have strong channel correlations [56]. As in [56], we can show that most images in the PASCAL VOC'07 dataset have weak channel correlations. However, traditional dictionary learning works well only when the conditions in the training set are similar as those present during testing [57], which implies that the learned dictionary is

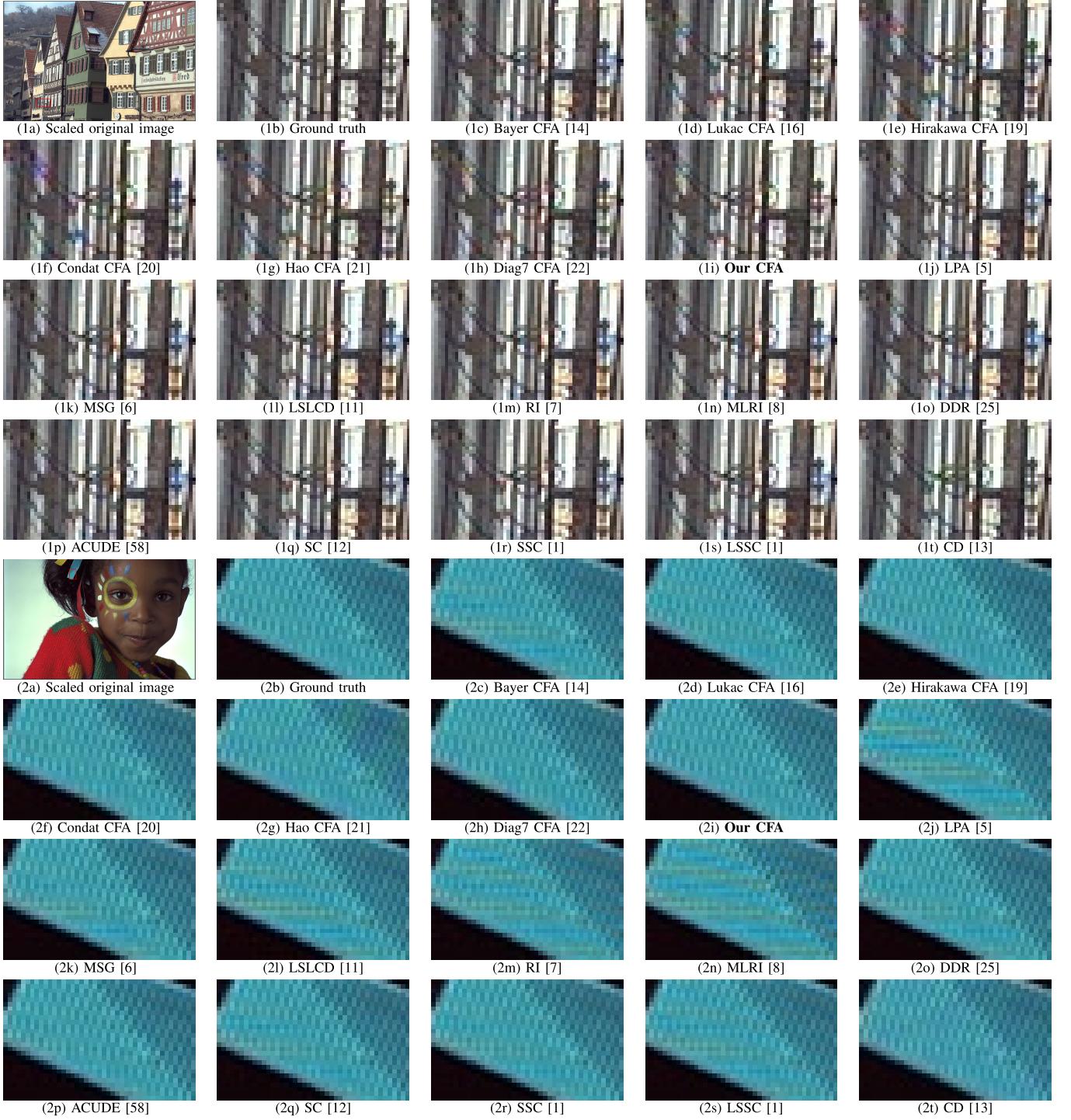


Fig. 4. Blowups of some demosaicked images in the Kodak dataset. From top to bottom, the images are from #8 and #15 Images of the Kodak dataset, respectively. In each group, (a) is the scaled original image, in which the red rectangle indicates the selected patch to blow up; (b) is the ground truth; (c)-(h) and (i) are the images demosaicked from raw images by the predefined CFAs and our specifically optimized CFA, respectively; (j)-(t) are the images demosaicked by the state-of-the-art demosaicking algorithms with their respective best CFAs. From the two groups of images, we can clearly see that there are obvious false color artifacts along edges in the images demosaicked from raw images by other CFAs, while those by our CFA have better visual quality.

optimal only for the Kodak dataset. Accordingly, we only test our CFA design method with the learned dictionary on the Kodak dataset, which contains 24 RGB images with a size of  $512 \times 768$  pixels.

2) *Compared CFAs and Demosaicking Algorithms:* As we assume that a CFA pattern should be smaller than an image

patch (here is of  $8 \times 8$  pixels), we choose the  $2 \times 2$  Bayer CFA [14] (Fig. 1(a)),  $4 \times 2$  Lukac CFA [16] (Fig. 1(b)),  $4 \times 2$  Hirakawa CFA [19] (Fig. 1(d)),  $3 \times 2$  Condat CFA [20] (Fig. 1(e)),  $4 \times 4$  Hao CFA [21] (Fig. 1(f)), and  $7 \times 7$  Diag7 CFA [22] (Fig. 1(g)) for comparison. They are all demosaicked by PSRD for fairness.

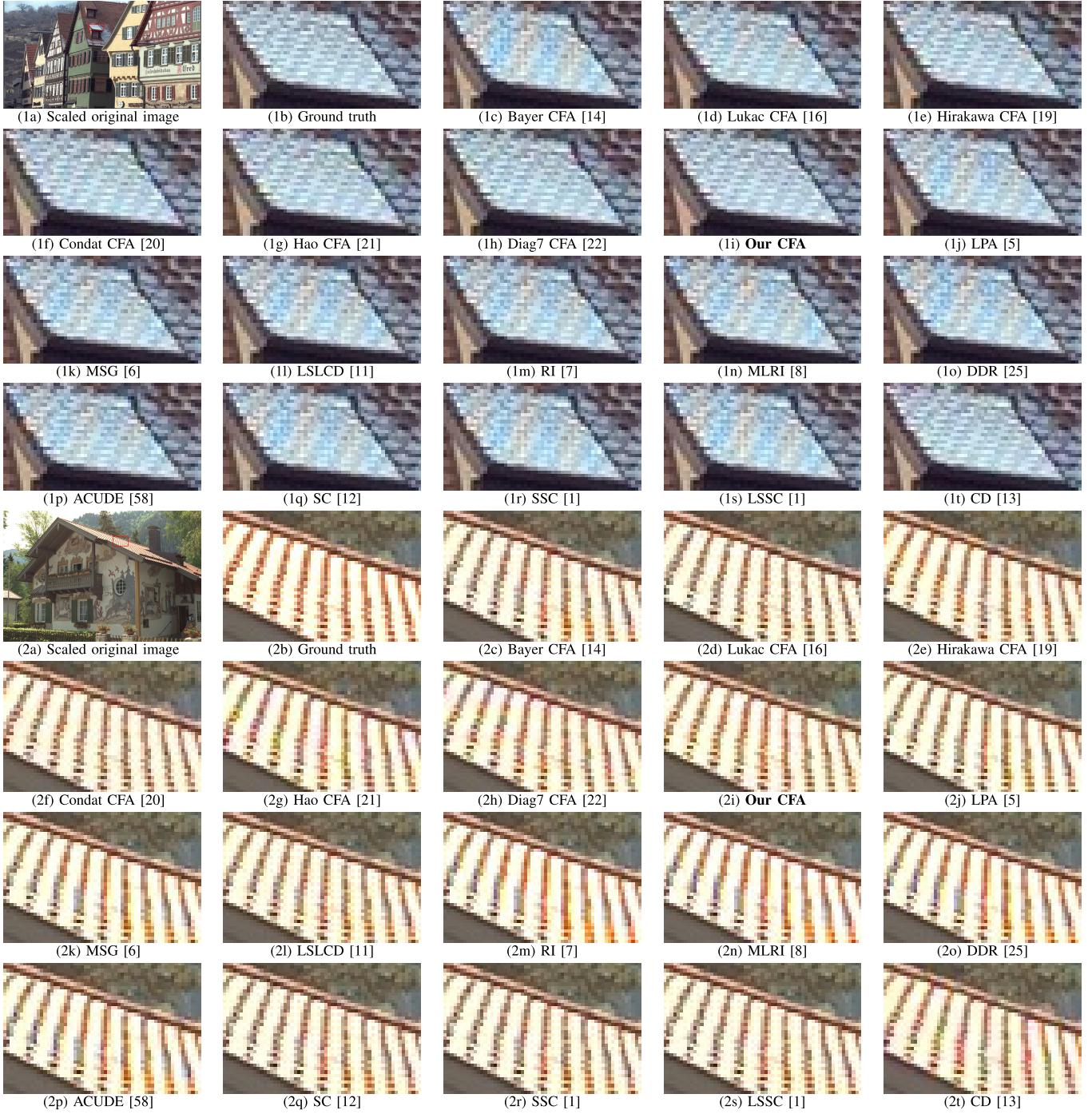


Fig. 5. More blowups of demosaicked images in the Kodak dataset. From top to bottom, the images are from #8 and #24 Images of the Kodak dataset, respectively. In each group, (a) is the scaled original image, in which the red rectangle indicates the selected patch to blow up; (b) is the ground truth; (c)-(h) and (i) are the images demosaicked from raw images by the predefined CFAs and our specifically optimized CFA, respectively; (j)-(t) are the images demosaicked by the state-of-the-art demosaicking algorithms with their respective best CFAs. From the two groups of images, we can see that the images demosaicked from the raw images by other CFAs have severe zipper effects or false color artifacts, while those by our CFA have better subjective quality.

To further demonstrate the benefit of our CFA design method in enhancing the performance of sparse representation based demosaicking, we also include comparisons between PSRD with our optimized CFA and state-of-the-art demosaicking algorithms with their respective best CFAs. The selected demosaicking algorithms all achieve their best performance with the Bayer CFA unless stated otherwise. LPA [5] designed

a spatially adaptive filter to remove demosaicking errors. MSG [6] used multiscale color gradients to adaptively combine directional estimates. LSLCD [11] is a frequency selection based demosaicking algorithm, which learns bandpass filters by minimizing the mean-squared demosaicking error over a training image set. It also employed an adaptive weighting strategy [10] to improve demosaicking performance. RI [7]

TABLE II

EVALUATION ON THE KODAK DATASET. THE LEFT PART SHOWS THE COMPARISON BETWEEN OUR SPECIFICALLY OPTIMIZED CFA AND SIX REPRESENTATIVE CFAS DEMOSAICKED ALL BY PSRD. THE RIGHT PART SHOWS THE COMPARISON BETWEEN PSRD WITH OUR OPTIMIZED CFA AND THE STATE-OF-THE-ART DEMOSAICKING ALGORITHMS WITH THEIR RESPECTIVE BEST CFAS. “ALG.” AND “AVG.” STAND FOR “ALGORITHM” AND “AVERAGE”, RESPECTIVELY. THE INDIVIDUAL AND AVERAGE CPSNR VALUES ARE REPORTED

CFA	Bayer [14]	Lukac [16]	Hirakawa [19]	Condat [20]	Hao [21]	Diag7 [22]	Ours
Alg.	PSRD						
01	40.51	39.56	39.88	39.91	40.24	40.85	<b>41.30</b>
02	41.90	41.26	42.17	42.04	42.50	42.63	<b>42.65</b>
03	43.20	42.67	43.25	43.77	43.36	<b>43.89</b>	43.52
04	42.20	40.95	41.67	42.07	<b>42.59</b>	41.77	42.22
05	38.88	37.43	38.88	38.74	39.27	<b>39.36</b>	39.14
06	41.20	40.87	41.57	41.52	41.71	42.02	<b>42.47</b>
07	43.30	42.51	43.38	<b>43.70</b>	43.45	43.18	43.61
08	37.44	37.44	38.23	38.22	38.49	38.27	<b>39.08</b>
09	43.13	42.35	42.96	43.29	43.35	43.53	<b>43.62</b>
10	42.82	42.32	43.12	43.34	43.48	<b>43.62</b>	43.55
11	41.22	40.78	41.30	41.39	41.67	41.32	<b>42.30</b>
12	44.26	43.63	44.36	44.67	44.60	44.91	<b>44.95</b>
13	36.00	35.09	34.95	35.12	35.17	35.40	<b>36.52</b>
14	37.93	37.57	38.39	38.69	38.35	38.09	<b>38.71</b>
15	41.00	39.87	40.77	40.92	41.29	40.60	<b>41.37</b>
16	44.35	44.28	44.64	44.96	44.64	44.91	<b>45.59</b>
17	41.80	41.00	41.65	41.59	42.13	41.79	<b>42.38</b>
18	38.07	37.23	37.56	37.65	37.70	37.62	<b>38.39</b>
19	41.65	41.15	41.80	41.75	42.03	42.15	<b>42.57</b>
20	41.75	41.23	42.18	42.05	42.33	42.42	<b>42.55</b>
21	40.47	40.17	40.62	40.69	40.73	40.55	<b>41.54</b>
22	38.83	38.22	39.14	39.23	39.31	<b>39.42</b>	39.35
23	43.48	43.05	43.86	44.13	43.95	43.88	<b>44.14</b>
24	35.50	35.29	36.49	36.35	36.56	<b>36.63</b>	36.60
Avg.	40.87	40.25	40.95	41.07	41.20	41.20	<b>41.59</b>
	40.52	41.02	40.63	39.97	40.20	41.06	40.78

	Bayer [14]										Hirakawa [19]	Ours
LPA	MSG	LSLCD	RI	MLRI	DDR	ACUDE	SC	SSC	LSSC	CD	PSRD	
[5]	[6]	[11]	[7]	[8]	[25]	[58]	[12]	[1]	[1]	[13]		
40.46	39.87	39.51	39.06	39.23	40.05	39.52	40.85	41.33	41.38	<b>41.54</b>	41.30	
41.32	41.77	41.55	40.06	40.93	41.48	41.42	41.75	41.95	42.19	41.29	<b>42.65</b>	
43.50	43.76	43.26	41.71	42.77	43.94	42.89	43.20	43.40	<b>44.21</b>	42.70	43.52	
40.89	41.44	41.19	40.35	40.81	41.62	41.18	42.24	42.47	<b>42.81</b>	40.95	42.22	
37.56	39.08	38.76	37.80	37.79	39.26	38.85	38.79	38.96	<b>39.46</b>	38.65	39.14	
40.95	41.41	40.98	40.90	40.59	41.38	41.22	41.34	41.69	41.74	41.79	<b>42.47</b>	
43.03	43.52	43.57	41.99	42.38	43.60	43.43	43.26	43.56	<b>43.96</b>	42.61	43.61	
37.14	37.58	36.09	36.59	36.68	37.47	37.80	37.47	37.60	37.58	38.81	<b>39.08</b>	
43.52	43.81	42.94	42.56	43.05	43.70	43.72	43.21	43.42	<b>43.91</b>	43.55	43.62	
42.67	43.24	42.82	42.47	42.54	43.35	43.17	42.89	42.99	43.24	42.92	<b>43.55</b>	
40.55	41.39	40.46	40.28	40.50	41.28	41.04	41.25	41.40	41.55	40.90	<b>42.30</b>	
44.00	44.48	43.71	43.30	43.74	44.54	44.20	44.28	44.62	44.88	44.61	<b>44.95</b>	
36.11	36.02	<b>36.56</b>	35.43	35.26	36.15	35.75	36.21	36.36	36.47	36.50	36.52	
37.01	38.02	37.87	36.34	37.00	38.16	37.24	37.78	37.95	<b>38.83</b>	36.69	38.71	
40.09	40.31	40.69	38.85	39.20	40.25	40.13	41.03	41.45	<b>41.76</b>	39.81	41.37	
44.01	44.89	44.37	44.30	44.14	44.64	44.80	44.36	44.84	44.90	44.82	<b>45.59</b>	
41.83	42.34	41.97	41.79	41.59	42.28	42.25	41.81	41.94	42.02	<b>42.46</b>	42.38	
37.36	38.19	37.83	37.52	37.45	38.32	38.08	38.13	38.23	<b>38.46</b>	38.30	38.39	
41.51	42.07	41.02	41.31	41.17	41.96	41.66	41.86	42.10	42.34	<b>42.68</b>	42.57	
41.43	42.15	41.41	41.23	41.38	42.32	42.01	41.92	41.96	42.19	42.41	<b>42.55</b>	
39.65	40.33	39.65	39.67	39.68	40.53	40.28	40.60	40.68	40.70	41.05	<b>41.54</b>	
38.52	39.08	38.92	38.50	38.78	<b>39.39</b>	38.98	38.81	39.03	39.33	39.15	39.35	
43.92	44.04	<b>44.28</b>	42.02	43.17	43.96	43.55	43.47	43.79	44.27	43.05	44.14	
35.35	35.67	35.70	35.25	34.95	35.76	35.61	35.53	35.53	35.81	<b>36.77</b>	36.60	
40.52	41.02	40.63	39.97	40.20	41.06	40.78	40.92	41.14	41.42	41.00	<b>41.59</b>	

performed the interpolation in a residual domain, where the residuals are differences between known and tentatively estimated color components. MLRI [8] is an extension of RI, which estimates the tentative color components by minimizing the Laplacian energies of the residuals. DDR [25] integrated MLRI and a fast postprocessing algorithm based on directional difference regression, where the regressors are offline learned on a training image set. ACUDE [58] was developed based on the accurate chrominance estimation in the spatial domain and the optimal demosaicking color transformation. SC [12] is similar to PSRD but used a new inner product in the standard OMP. SSC [1] is a simultaneous sparse representation based demosaicking algorithm, which promotes similar raw image patches to have similar sparse representations. LSSC [1] is an extension of SSC, which further learns an adaptive dictionary for every test image. It is important to note that we use the identical dictionary (see IV-A.1) as SC and SSC do, which is also the initial dictionary for LSSC. CD [13] is a sparse representation based demosaicking algorithm with a manually constructed dictionary. It achieved the best performance with the Hirakawa CFA and an adaptive block selection strategy.

### B. Comparison With the Predefined CFAs

For the given dictionary [53] and a given CFA pattern size, we generate a CFA pattern by solving our CFA design model in (5). We check all pattern sizes that are smaller than  $8 \times 8$ . The resulted CFA pattern that performs the best in our tests is shown in Fig. 3. It is a  $4 \times 2$  CFA pattern and has seven different color components.

We compare our specifically optimized CFA with six representative CFAs on the Kodak dataset. The left part of Table II gives the individual and average CPSNR values, where the best values are in boldface. We can see that our optimized CFA achieves the best CPSNR values on most individual images and the whole dataset.

We also present part of the visual comparison on the Kodak dataset in Fig. 4 and Fig. 5. We can see that the visual quality of our specifically optimized CFA is better than that of the predefined CFAs (Please read the captions for the descriptions on visual difference.).

The two comparisons validate the advantage of our optimized CFA and also testify that designing CFAs specifically for a type of demosaicking algorithms is important for achieving superior performance.

### C. Comparison With the State-of-the-Art Demosaicking

Both the CFA and the demosaicking algorithm affect the quality of the reconstructed color images. Accordingly, to further demonstrate the benefit of our CFA design method in improving the performance of sparse representation based demosaicking, we also provide the comparison between PSRD with our specifically optimized CFA and the state-of-the-art demosaicking algorithms with their respective best CFAs. The right part of Table II presents the individual and average CPSNR values on the Kodak dataset, where the best values are in boldface. We can see that LSSC [1] and PSRD with our optimized CFA outperform other demosaicking algorithms on most individual images and the whole dataset. Moreover,

PSRD with our optimized CFA achieves the best average CPSNR value on the Kodak dataset, which is slightly better than LSSC with a gap of 0.17dB. *To the best of our knowledge, it is the first sparse representation based demosaicking algorithm that outperforms LSSC on the Kodak dataset in CPSNR since it was proposed in 2009.* Nonetheless, it is important to highlight that LSSC first obtains a tentatively demosaicked image using SC [12] with an initial dictionary. Then it learns a new dictionary on the tentatively demosaicked image. It next concatenates the two dictionaries and performs SSC [1] with the concatenated dictionary to get the finally demosaicked image. So LSSC has two complete demosaicking processes and one dictionary learning process. However, PSRD with our optimized CFA uses a fixed dictionary and has only one complete demosaicking process. When we compare with the sparse representation based demosaicking with a fixed dictionary on the whole dataset, PSRD with our optimized CFA performs better than SC [12], SSC [1], and CD [13] with gaps of 0.67dB, 0.45dB, and 0.59dB, respectively, which are significant.

We present part of the visual comparison in Fig. 4 and Fig. 5. We can see that PSRD with our optimized CFA produces better visual quality than that of other demosaicking algorithms with their respective best CFAs (Please read the captions for the descriptions on visual difference.).

The above two comparisons further verify that specifically optimizing CFAs for a type of demosaicking algorithms is beneficial in producing high quality demosaicked images.

## V. CONCLUSIONS

In this paper, we present a theoretically grounded approach to design specific CFAs for sparse representation based demosaicking, where the dictionary is well-chosen and fixed. We formulate the CFA design as the minimization of the mutual coherence with the CFA's physical realizability constraints, where most methods for minimizing the mutual coherence do not apply. We develop a new method to solve it based on generalized fractional programming. Extensive experiments on benchmark images demonstrate the superiority of our design method. Future work will include learning channel-correlation adaptive dictionaries [57], investigating the effect of CFA pattern size, and jointly designing the CFA and dictionary as did for the CS projection matrix [37], [59].

## APPENDIX SOLVING PROBLEM (18)

Substituting (17) into (18), we rewrite problem (18) in more detail:

$$\min_{\mathbf{P} \in \Omega} \max_{0 \leq l \leq t} \left\{ \langle \mathbf{G}^l + \eta(\mathbf{P}^l - \hat{\mathbf{P}}^l), \mathbf{P} - \hat{\mathbf{P}}^l \rangle - \hat{e}_l^t \right\} + \frac{\nu}{2} \|\mathbf{P} - \hat{\mathbf{P}}^l\|_F^2, \quad (23)$$

where the term  $f(\hat{\mathbf{P}}^l)$  of  $\varphi^l(\mathbf{P})$  in (17) is omitted as this does not affect the solution. It can be reformulated as:

$$\min_{\mathbf{p} \in \Psi} \max_{0 \leq l \leq t} \left\{ \langle \mathbf{s}_l^t, \mathbf{p} \rangle - \hat{e}_l^t \right\} + \frac{\nu}{2} \|\mathbf{p} - \hat{\mathbf{p}}^l\|_2^2, \quad (24)$$

where  $\Psi = \{\mathbf{p} | \mathbf{p} = \text{vec}(\mathbf{P}), \mathbf{P} \in \Omega\}$ ,  $\mathbf{s}_l^t = \text{vec}(\mathbf{G}^l + \eta(\mathbf{P}^l - \hat{\mathbf{P}}^l))$ ,  $\mathbf{p} = \text{vec}(\mathbf{P})$ ,  $\hat{\mathbf{p}}^l = \text{vec}(\hat{\mathbf{P}}^l)$ , and  $\hat{e}_l^t = \hat{e}_l^t + \langle \mathbf{s}_l^t, \hat{\mathbf{p}}^l \rangle$ . Denote  $\mathbf{S}^t = (\mathbf{s}_0^t, \dots, \mathbf{s}_t^t)$  and  $\hat{\mathbf{e}}^t = (\hat{e}_0^t, \dots, \hat{e}_t^t)^T$ , then problem (24) can be reformulated as:

$$\begin{aligned} & \min_{\mathbf{p} \in \Psi, z} z + \frac{\nu}{2} \|\mathbf{p} - \hat{\mathbf{p}}^l\|_2^2 \\ & \text{s.t. } z\mathbf{1} - \mathbf{S}^{t^T} \mathbf{p} + \hat{\mathbf{e}}^t \geq \mathbf{0}, \end{aligned} \quad (25)$$

where  $\mathbf{1} \in \mathbb{R}^{(t+1) \times 1}$ . We introduce an auxiliary variable  $\mathbf{k}$  to equivalently rewrite problem (25) as follows:

$$\begin{aligned} & \min_{\mathbf{p}, z, \mathbf{k}} z + \frac{\nu}{2} \|\mathbf{p} - \hat{\mathbf{p}}^l\|_2^2 + \mathcal{I}_{\Psi}(\mathbf{p}) + \mathcal{I}_{\mathbb{R}_+}(\mathbf{k}) \\ & \text{s.t. } z\mathbf{1} - \mathbf{S}^{t^T} \mathbf{p} + \hat{\mathbf{e}}^t - \mathbf{k} = \mathbf{0}, \end{aligned} \quad (26)$$

where  $\mathcal{I}_{\Psi}(\cdot)$  and  $\mathcal{I}_{\mathbb{R}_+}(\cdot)$  are indicator functions defined on  $\Psi$  and  $\mathbb{R}_+$ , respectively. The indicator function of a set  $\Phi$  is defined as

$$\mathcal{I}_{\Phi}(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \in \Phi, \\ +\infty, & \text{otherwise.} \end{cases}$$

Problem (26) is a three-block separable convex problem with linear constraints. In order to avoid expensive matrix inversions, we use the linearized alternating direction method with parallel splitting and adaptive penalty (LADMPSAP) [60] to solve it. The convergence of LADMPSAP for solving multi-block separable convex problems is well established.

The augmented Lagrangian function of problem (26) is:

$$\begin{aligned} \mathcal{L}(\mathbf{p}, z, \mathbf{k}, \mathbf{y}) = & z + \frac{\nu}{2} \|\mathbf{p} - \hat{\mathbf{p}}^l\|_2^2 + \mathcal{I}_{\Psi}(\mathbf{p}) + \mathcal{I}_{\mathbb{R}_+}(\mathbf{k}) \\ & + \langle \mathbf{y}, z\mathbf{1} - \mathbf{S}^{t^T} \mathbf{p} + \hat{\mathbf{e}}^t - \mathbf{k} \rangle \\ & + \frac{\beta}{2} \|z\mathbf{1} - \mathbf{S}^{t^T} \mathbf{p} + \hat{\mathbf{e}}^t - \mathbf{k}\|_2^2, \end{aligned} \quad (27)$$

where  $\mathbf{y}$  is the Lagrange multiplier and  $\beta > 0$  is the penalty parameter which is updated during the iteration.

Then by LADMPSAP, problem (26) can be solved via the following iterations:

$$\hat{\mathbf{y}}^\tau = \mathbf{y}^\tau + \beta(z^\tau \mathbf{1} - \mathbf{S}^{t^T} \mathbf{p}^\tau + \hat{\mathbf{e}}^\tau - \mathbf{k}^\tau), \quad (28)$$

$$\begin{aligned} \mathbf{p}^{\tau+1} = & \underset{\mathbf{p}}{\operatorname{argmin}} \mathcal{I}_{\Psi}(\mathbf{p}) + \frac{\theta\sigma}{2} \|\mathbf{p} - \mathbf{p}^\tau + \nabla \mathcal{G}(\mathbf{p}^\tau)/(\theta\sigma)\|_2^2 \\ = & \pi_{\Psi}(\mathbf{p}^\tau - \nabla \mathcal{G}(\mathbf{p}^\tau)/(\theta\sigma)), \end{aligned} \quad (29)$$

$$\begin{aligned} z^{\tau+1} = & \underset{z}{\operatorname{argmin}} \frac{\theta\beta(t+1)}{2} \|z - z^\tau + \nabla \mathcal{F}(z^\tau)/(\theta\beta(t+1))\|_2^2 \\ = & z^\tau - \nabla \mathcal{F}(z^\tau)/(\theta\beta(t+1)), \end{aligned} \quad (30)$$

$$\begin{aligned} \mathbf{k}^{\tau+1} = & \underset{\mathbf{k}}{\operatorname{argmin}} \mathcal{I}_{\mathbb{R}_+}(\mathbf{k}) + \frac{\theta\beta}{2} \|\mathbf{k} - \mathbf{k}^\tau + \nabla \mathcal{Q}(\mathbf{k}^\tau)/(\theta\beta)\|_2^2 \\ = & \max(0, \mathbf{k}^\tau - \nabla \mathcal{Q}(\mathbf{k}^\tau)/(\theta\beta)), \end{aligned} \quad (31)$$

$$\mathbf{y}^{\tau+1} = \mathbf{y}^\tau + \beta(z^{\tau+1} \mathbf{1} - \mathbf{S}^{t^T} \mathbf{p}^{\tau+1} + \hat{\mathbf{e}}^\tau - \mathbf{k}^{\tau+1}), \quad (32)$$

where  $\theta$  is a constant that should be larger than or equal to the number of blocks in (26),  $\mathcal{G}(\mathbf{p}) = \frac{\nu}{2} \|\mathbf{p} - \hat{\mathbf{p}}^l\|_2^2 + \frac{\beta}{2} \|z^\tau \mathbf{1} - \mathbf{S}^{t^T} \mathbf{p} + \hat{\mathbf{e}}^\tau - \mathbf{k}^\tau + \hat{\mathbf{y}}^\tau/\beta\|_2^2$  in (29),  $\nabla \mathcal{G}(\mathbf{p}) = \nu(\mathbf{p} - \hat{\mathbf{p}}^l) - \beta \mathbf{S}^{t^T} (z^\tau \mathbf{1} - \mathbf{S}^{t^T} \mathbf{p} + \hat{\mathbf{e}}^\tau - \mathbf{k}^\tau + \hat{\mathbf{y}}^\tau/\beta)$  is the derivative of  $\mathcal{G}(\mathbf{p})$ ,  $\sigma$  is the largest eigenvalue of  $\nu \mathbf{I} + \beta \mathbf{S}^{t^T} \mathbf{S}^T$ ,  $\pi_{\Psi}(\mathbf{u})$  is a linear operator that

**Algorithm 4** Algorithm for Solving Problem (26)

**Input:**  $\mathbf{p}^t, \hat{\mathbf{p}}^t, \mathbf{S}^t, \hat{\mathbf{e}}^t, \nu, m, r \times c, \theta = 3, \beta = 1, \beta_{\max} = 10^5, v = 2, \varepsilon_1 = 10^{-6}, \varepsilon_2 = 10^{-8}$ .

- 1: **Initialization:**  $\mathbf{p} = \mathbf{p}^t, z = 0, \mathbf{k} = \mathbf{0}, \mathbf{y} = \mathbf{0}, \tau = 0$ .
- 2: **while** the stop conditions (34) and (35) are not met **do**
- 3:   compute  $\hat{\mathbf{y}}$  as (28).
- 4:   update  $\mathbf{p}, z$ , and  $\mathbf{k}$  by (29), (30), and (31), respectively.
- 5:   update  $\mathbf{y}$  by (32) and  $\beta$  by (33).
- 6:    $\tau \leftarrow \tau + 1$ .
- 7: **end while**

**Output:**  $\mathbf{p}$ .

---

projects  $\mathbf{u}$  onto  $\Psi$ ,  $\mathcal{F}(z) = z + \frac{\beta}{2} \|z\mathbf{1} - \mathbf{S}^T \mathbf{p}^\tau + \hat{\mathbf{e}}^\tau - \mathbf{k}^\tau + \hat{\mathbf{y}}^\tau / \beta\|^2$  in (30),  $\nabla \mathcal{F}(z) = 1 + \beta \mathbf{1}^T (z\mathbf{1} - \mathbf{S}^T \mathbf{p}^\tau + \hat{\mathbf{e}}^\tau - \mathbf{k}^\tau + \hat{\mathbf{y}}^\tau / \beta)$  is the derivative of  $\mathcal{F}(z)$ ,  $\mathcal{Q}(\mathbf{k}) = \frac{\beta}{2} \|z\mathbf{1} - \mathbf{S}^T \mathbf{p}^\tau + \hat{\mathbf{e}}^\tau - \mathbf{k} + \hat{\mathbf{y}}^\tau / \beta\|^2$  in (31), and  $\nabla \mathcal{Q}(\mathbf{k}) = -\beta(z\mathbf{1} - \mathbf{S}^T \mathbf{p}^\tau + \hat{\mathbf{e}}^\tau - \mathbf{k} + \hat{\mathbf{y}}^\tau / \beta)$ . It should be noted that  $\mathbf{p}, z$ , and  $\mathbf{k}$  are updated in parallel.

The computation of  $\pi_\Psi(\mathbf{u})$  is as follows. We first convert  $\mathbf{u}$  into a matrix  $\mathbf{U} \in \mathbb{R}^{m \times 3m}$  satisfying  $\text{vec}(\mathbf{U}) = \mathbf{u}$ . Then we form  $\mathbf{C} \in \mathbb{R}^{\sqrt{m} \times \sqrt{m} \times 3}$ , which satisfies  $\text{vec}(\mathbf{C}(:, :, 1)) = \text{diag}(\mathbf{U}(:, 1 : m))$ ,  $\text{vec}(\mathbf{C}(:, :, 2)) = \text{diag}(\mathbf{U}(:, m + 1 : 2m))$ , and  $\text{vec}(\mathbf{C}(:, :, 3)) = \text{diag}(\mathbf{U}(:, 2m + 1 : 3m))$ . We next normalize  $\mathbf{C}$  to obtain  $\tilde{\mathbf{C}}$  such that  $\tilde{\mathbf{C}} \geq \mathbf{0}$  and  $\sum_i \tilde{\mathbf{C}}(:, :, i) = \mathbf{1}$ . We average all nonoverlapping  $r \times c$  blocks of  $\tilde{\mathbf{C}}$  to produce a valid CFA pattern. With the CFA pattern, we generate a periodic CFA  $\tilde{\mathbf{C}} \in \mathbb{R}^{\sqrt{m} \times \sqrt{m} \times 3}$  and form the tentative mosaicking matrix  $\tilde{\mathbf{U}} = (\tilde{\mathbf{U}}_R, \tilde{\mathbf{U}}_G, \tilde{\mathbf{U}}_B)$ , where  $\tilde{\mathbf{U}}_R = \text{Diag}(\text{vec}(\tilde{\mathbf{C}}(:, :, 1)))$ ,  $\tilde{\mathbf{U}}_G = \text{Diag}(\text{vec}(\tilde{\mathbf{C}}(:, :, 2)))$ , and  $\tilde{\mathbf{U}}_B = \text{Diag}(\text{vec}(\tilde{\mathbf{C}}(:, :, 3)))$ . We finally compute  $\pi_\Psi(\mathbf{u})$  by  $\pi_\Psi(\mathbf{u}) = \text{vec}(\tilde{\mathbf{U}})$ .

We update  $\beta$  adaptively by:

$$\beta^{\tau+1} = \begin{cases} \min(\beta_{\max}, \beta^\tau v), & \text{if } \beta^\tau \alpha^\tau \leq \varepsilon_1, \\ \beta^\tau, & \text{otherwise,} \end{cases} \quad (33)$$

where  $\beta_{\max}$  is an upper bound of  $\{\beta^\tau\}$ ,  $v \geq 1$  is a constant, and  $\alpha^\tau = \max\{\sqrt{\sigma/\beta} \|\mathbf{p}^{\tau+1} - \mathbf{p}^\tau\|_2, \sqrt{t+1} |z^{\tau+1} - z^\tau|, \|\mathbf{k}^{\tau+1} - \mathbf{k}^\tau\|_2\}$ . If  $\beta$  is changed, we recalculate  $\sigma$  as it is the largest eigenvalue of  $\nu \mathbf{I} + \beta \mathbf{S}^T \mathbf{S}^T$ . The stopping criteria are:

$$\beta^\tau \alpha^\tau \leq \varepsilon_1 \text{ and} \quad (34)$$

$$\|z^{\tau+1} \mathbf{1} - \mathbf{S}^T \mathbf{p}^{\tau+1} + \hat{\mathbf{e}}^\tau - \mathbf{k}^{\tau+1}\|_2 \leq \varepsilon_2. \quad (35)$$

We summarize the whole solution process of problem (26) in Algorithm 4.

## REFERENCES

- [1] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. IEEE 12th Int. Conf. Comput. Vis.* Oct. 2009, pp. 2272–2279.
- [2] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosacking: Color filter array interpolation," *IEEE Signal Process. Mag.*, vol. 22, no. 1, pp. 44–54, Jan. 2005.
- [3] X. Li, B. Gunturk, and L. Zhang, "Image demosaicing: A systematic survey," *Proc. SPIE*, vol. 6822, p. 68221J, Jan. 2008.
- [4] D. Menon and G. Calvagno, "Color image demosaicing: An overview," *Signal Process., Image Commun.*, vol. 26, nos. 8–9, pp. 518–533, Oct. 2011.
- [5] D. Paliy, V. Katkovnik, R. Bilcu, S. Alenius, and K. Egiazarian, "Spatially adaptive color filter array interpolation for noiseless and noisy data," *Int. J. Imag. Syst. Technol.*, vol. 17, no. 3, pp. 105–122, 2007.
- [6] I. Pekkucuksen and Y. Altunbasak, "Multiscale gradients-based color filter array interpolation," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 157–165, Jan. 2013.
- [7] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Residual interpolation for color image demosaicing," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 2304–2308.
- [8] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Minimized-laplacian residual interpolation for color image demosaicing," *Proc. SPIE*, vol. 9023, p. 90230L, Mar. 2014.
- [9] D. Alleysson, S. Susstrunk, and J. Herault, "Linear demosaicing inspired by the human visual system," *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 439–449, Apr. 2005.
- [10] E. Dubois, "Frequency-domain methods for demosaicing of Bayer-sampled color images," *IEEE Signal Process. Lett.*, vol. 12, no. 12, pp. 847–850, Dec. 2005.
- [11] B. Leung, G. Jeon, and E. Dubois, "Least-squares luma-chroma demultiplexing algorithm for Bayer demosaicing," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 1885–1894, Jul. 2011.
- [12] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.
- [13] A. A. Moghadam, M. Aghagolzadeh, M. Kumar, and H. Radha, "Compressive framework for demosaicing of natural images," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2356–2371, Jun. 2013.
- [14] B. E. Bayer, "Color imaging array," U.S. Patent 3 971 065, Jul. 20, 1976.
- [15] (Dec. 1, 2013). *Realization of Natural Color Reproduction in Digital Still Cameras. Closer to the Natural Sight Perception of the Human Eye*. [Online]. Available: <http://www.sony.net/SonyInfo/News/Press/200307/03-029E/>
- [16] R. Lukac and K. N. Plataniotis, "Color filter arrays: Design and performance analysis," *IEEE Trans. Consum. Electron.*, vol. 51, no. 4, pp. 1260–1267, Nov. 2005.
- [17] J. T. Compton and J. F. Hamilton, Jr., "Image sensor with improved light sensitivity," U.S. Patent 2007 0024931, Feb. 1, 2007.
- [18] (Nov. 10, 2014). *X-Trans CMOS*. [Online]. Available: [http://www.fujifilmusa.com/products/digital\\_cameras/x/fujifilm\\_x\\_pro1/](http://www.fujifilmusa.com/products/digital_cameras/x/fujifilm_x_pro1/)
- [19] K. Hirakawa and P. J. Wolfe, "Spatio-spectral color filter array design for optimal image recovery," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1876–1890, Oct. 2008.
- [20] L. Condat, "A new color filter array with optimal properties for noiseless and noisy color image acquisition," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2200–2210, Aug. 2011.
- [21] P. Hao, Y. Li, Z. Lin, and E. Dubois, "A geometric method for optimal design of color filter arrays," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 709–722, Mar. 2011.
- [22] C. Bai, J. Li, Z. Lin, and J. Yu, "Automatic design of color filter arrays in the frequency domain," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1793–1807, Apr. 2016.
- [23] A. Chakrabarti, W. T. Freeman, and T. Zickler, "Rethinking color cameras," in *Proc. IEEE Int. Conf. Comput. Photogr.*, May 2014, pp. 1–8.
- [24] D. Gao, X. Wu, G. Shi, and L. Zhang, "Color demosaicing with an image formation model and adaptive PCA," *J. Vis. Commun. Image Represent.*, vol. 23, no. 7, pp. 1019–1030, Oct. 2012.
- [25] J. Wu, R. Timofte, and L. Van Gool, "Demosaicing based on directional difference regression and efficient regression priors," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3862–3874, Jun. 2016.
- [26] J. Wang, C. Zhang, and P. Hao, "New color filter arrays of high light sensitivity and high demosaicing performance," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 3153–3156.
- [27] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
- [28] Y. C. Pati, R. Rezaifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. Conf. Rec. 27th Asilomar Conf. Signals, Syst. Comput.*, Nov. 1993, pp. 40–44.
- [29] M. F. Duarte *et al.*, "Single-pixel imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 83–91, Mar. 2008.
- [30] P. Nagesh and B. Li, "Compressive imaging of color images," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2009, pp. 1261–1264.

- [31] A. Majumdar and R. K. Ward, "Compressed sensing of color images," *Signal Process.*, vol. 90, no. 12, pp. 3122–3127, Dec. 2010.
- [32] (Dec. 1, 2013). *Kodak Lossless True Color Image Suite*. [Online]. Available: <http://r0k.us/graphics/kodak/>
- [33] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.
- [34] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
- [35] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Trans. Inf. Theory*, vol. 49, no. 12, pp. 3320–3325, Dec. 2003.
- [36] M. Elad, "Optimized projections for compressed sensing," *IEEE Trans. Signal Process.*, vol. 55, no. 12, pp. 5695–5702, Dec. 2007.
- [37] J. M. Duarte-Carvajalino and G. Sapiro, "Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1395–1408, Jul. 2009.
- [38] J. Xu, Y. Pi, and Z. Cao, "Optimized projection matrix for compressive sensing," *EURASIP J. Adv. Signal Process.*, vol. 2010, no. 1, pp. 1–8, Dec. 2010.
- [39] N. Cleju, "Optimized projections for compressed sensing via rank-constrained nearest correlation matrix," *Appl. Comput. Harmon. Anal.*, vol. 36, no. 3, pp. 495–507, May 2014.
- [40] Z. Lin, C. Lu, and H. Li. (Aug. 2015). "Optimized projections for compressed sensing via direct mutual coherence minimization." [Online]. Available: <https://arxiv.org/abs/1508.03117>
- [41] C. Bai, J. Li, Z. Lin, J. Yu, and Y.-W. Chen, "Penrose demosaicking," *IEEE Trans. Image Process.*, vol. 24, no. 5, pp. 1672–1684, May 2015.
- [42] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vis. Res.*, vol. 37, no. 23, pp. 3311–3325, 1997.
- [43] J. P. Crouzeix, J. A. Ferland, and S. Schaible, "An algorithm for generalized fractional programs," *J. Optim. Theory Appl.*, vol. 47, no. 1, pp. 35–49, Sep. 1985.
- [44] J. P. Crouzeix, J. A. Ferland, and S. Schaible, "A note on an algorithm for generalized fractional programs," *J. Optim. Theory Appl.*, vol. 50, no. 1, pp. 183–187, Jul. 1986.
- [45] S. I. Birbil, J. B. G. Frenk, and S. Zhang, "Generalized fractional programming with user interaction," ERIM, Tech. Rep. ERS-2004-033-LIS, Rotterdam, the Netherlands, Mar. 2004.
- [46] J.-J. Strodiot, J.-P. Crouzeix, J. A. Ferland, and V. H. Nguyen, "An inexact proximal point method for solving generalized fractional programs," *J. Global Optim.*, vol. 42, no. 1, pp. 121–138, Sep. 2008.
- [47] K. C. Kiwiel, "Proximity control in bundle methods for convex nondifferentiable minimization," *Math. Program.*, vol. 46, no. 1, pp. 105–122, Jan. 1990.
- [48] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, vol. 317. Berlin, Germany: Springer-Verlag, 1998.
- [49] W. Hare and C. Sagastizábal, "A redistributed proximal bundle method for nonconvex optimization," *SIAM J. Optim.*, vol. 20, no. 5, pp. 2442–2473, 2010.
- [50] J. E. Kelley, Jr., "The cutting-plane method for solving convex programs," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 4, pp. 703–712, 1960.
- [51] K. C. Kiwiel, "A method for solving certain quadratic programming problems arising in nonsmooth optimization," *IMA J. Numer. Anal.*, vol. 6, no. 2, pp. 137–152, Apr. 1986.
- [52] (May 10, 2016). *Sparse Modeling Software*. [Online]. Available: <http://spams-devel.gforge.inria.fr/>
- [53] (Oct. 6, 2014). *Julien Mairal's Homepage*. [Online]. Available: <http://lear.inrialpes.fr/people/mairal/software.php>
- [54] M. Everingham, G. L. Van, C. K. I. Williams, J. Winn, and A. Zisserman. (Feb. 9, 2017). *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*. [Online]. Available: <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.htm>
- [55] L. Zhang, X. Wu, A. Buades, and X. Li, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *J. Electron. Imag.*, vol. 20, no. 2, p. 023016, Jun. 2011.
- [56] J. Duran and A. Buades, "Self-similarity and spectral correlation adaptive algorithm for color demosaicking," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 4031–4040, Sep. 2014.
- [57] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa, "Domain adaptive dictionary learning," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 631–645.
- [58] C. Zhang, Y. Li, J. Wang, and P. Hao, "Universal demosaicking of color filter arrays," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5173–5186, Nov. 2016.
- [59] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Apr. 2012.
- [60] Z. Lin, R. Liu, and H. Li, "Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning," *Mach. Learn.*, vol. 99, no. 2, pp. 287–325, 2015.

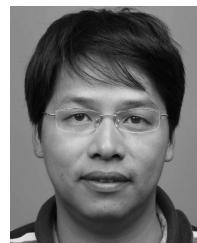
**Jia Li** received the B.S. degree in mathematics and M.S. degree in computer science from Zhengzhou University, Zhengzhou, China, in 2007 and 2012, respectively. He is currently pursuing the Ph.D. degree with the Beijing Key Laboratory of Traffic Data Analysis and Mining, School of Computer and Information Technology, Beijing Jiaotong University. His research interest is image processing.



**Chenyang Bai** received the B.E. and M.S. degrees from Hebei University, Baoding, China, in 2008 and 2011, respectively, and the Ph.D. degree from Beijing Jiaotong University, Beijing, China, in 2016, all in computer science. She is currently an Assistant Professor with the College of Information Engineering and Beijing Advanced Innovation Center for Imaging Technology, Capital Normal University, Beijing, China. Her research interests include image processing and machine learning.



**Zhouchen Lin** (M'00–SM'08) received the Ph.D. degree in applied mathematics from Peking University in 2000. He was a Chair Professor with Northeast Normal University. Currently, he is a Professor with the Key Laboratory of Machine Perception (Ministry of Education), School of Electronics Engineering and Computer Science, Peking University. His research interests include computer vision, image processing, machine learning, pattern recognition, and numerical optimization. He is an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and *International Journal of Computer Vision*. He is an IAPR Fellow.



**Jian Yu** received the B.S. and M.S. degrees in mathematics and Ph.D. degree in applied mathematics from Peking University, Beijing, China, in 1991, 1994, and 2000, respectively. He is currently a Professor with the School of Computer and Information Technology and the Director of the Beijing Key Laboratory of Traffic Data Analysis and Mining, Beijing Jiaotong University. His research interests include machine learning, image processing, and pattern recognition.

