Penrose Demosaicking

Chenyan Bai, Jia Li, Zhouchen Lin, *Senior Member, IEEE*, Jian Yu, and Yen-Wei Chen

***Abstract*— The Penrose pixel layout, an aperiodic pixel layout in rhombus Penrose tiling, has been shown to substantially outperform the existing square pixel layout in super-resolution. However, it was tested only on grayscale images. To study its performance on color images, we have to reconstruct regular color images from Penrose raw images, i.e., images with only one color component at each Penrose pixel, resulting in the problem of demosaicking from Penrose pixels. Penrose demosaicking is more difficult than regular demosaicking, because none of the color components of the reconstructed regular color images are available. Therefore, most of the traditional demosaicking methods do not apply. We develop a sparse representation-based method for Penrose demosaicking. Extensive experiments show that Penrose pixel layout outperforms regular pixel layouts in terms of both perceptual evaluation and S-CIELAB. The Penrose pixel layout is unique among all irregular layouts because it is uniformly three-colorable and it has only two pixel shapes, thick and thin rhombi, making its manufacturing relatively easy.**

***Index Terms*— Penrose, demosaicking, color filter array, sparse representation, alternating direction method (ADM).**

I. Introduction

MOST of the existing imaging sensors use the square pixel layout (square layout for short), i.e., periodic tiling of square pixels. Only a few literatures study the non-square pixel layout, which is mostly periodic tiling of hexagonal pixels [1]. For example, edge detection [2] and anti-aliasing [3] on the hexagonal pixel layout (hexagonal layout for short). Recently, a novel Penrose pixel layout (Penrose layout for short) [4], which uses pixels in rhombus Penrose tiling, was proposed for super-resolution [5].

Penrose tiling is an aperiodic tiling [4]. It is a covering of the plane by tiles with no overlaps or gaps. Rhombus Penrose tiling uses only two kinds of rhombi, having equal sides but different angles, which we call thick and thin rhombi (Fig. 1(c)). The ratio of areas of thick to thin rhombi is the golden number $\phi = \frac{1+\sqrt{5}}{2}$, which is also the ratio of their numbers [4].

The Penrose layout has been proven to be more effective in super-resolution than the square one, by both theory and simulation [5]. The routing scheme of wires for reading out Penrose pixel values has also been discussed in detail [5]. However, the Penrose layout was only tested on grayscale images. It is unclear whether we can obtain better color images from Penrose layout. Note that in most image capturing devices, every pixel is covered by a color filter. So every pixel has only one color component, e.g., red, green, or blue (R, G, or B) value (Fig. 1(b) and 1(d)). Such images are called raw images. The two missing color components at every pixel have to be estimated by the demosaicking technology (Fig. 1(f) and 1(g)). Demosaicking a Penrose raw image, or Penrose demosaicking for short, is more challenging than the regular demosaicking because none of the color components of the pixels of the reconstructed full color image is available (Fig. 1(c) and 1(e)) and the reconstructed color images are expected to be in square layout. As a result, most of the existing demosaicking methods, see [6]–[8], do not apply.

Another important issue is whether Penrose layout fits for reconstructing high quality color images. This requires Penrose layout to be uniformly three-colorable. Namely, each of the pixels can be assigned with one of the three color components and no adjacent pixels, i.e., pixels sharing a side, are assigned the same color component. Moreover, the numbers of pixels in the three color components are roughly the same. Uniform three-colorability ensures that all the color components can be densely sampled and the full color image can be recovered.[1] Fortunately, it is proven that Penrose layout is three-colorable [9]. Although there is no rigorous proof of uniform coloring yet, our empirical tests show that it is true. In contrast, many existing square layouts are not uniformly colored. For example, the Bayer color filter array (CFA) [10] has twice green components as red and blue ones (Fig. 1(b)). Finally, Penrose layout consists of only two pixel shapes [4], making the manufacturing of sensors in Penrose layout relatively easy (although not as easy as square and hexagonal layouts). All the above characteristics make Penrose layout unique among all irregular layouts.

[1]Recall that representing all color requires at least three color components, such as R, G, and B, or cyan, magenta, and yellow (C, M, and Y).

Manuscript received July 6, 2014; revised November 3, 2014 and February 8, 2015; accepted February 18, 2015. Date of publication March 6, 2015; date of current version March 23, 2015. The work of Z. Lin was supported in part by the 973 Program of China under Grant 2015CB352502, in part by the National Natural Science Foundation of China (NNSFC) under Grants 61272341 and 61231002, in part by the Cooperative Medianet Innovation Center, and in part by the Microsoft Research Asia Collaborative Research Program. The work of J. Yu was supported in part by the NNSFC under Grant 61033013, in part by the Ph.D. Programs Foundation of Ministry of Education of China under Grant 20120009110006, in part by the Program for Changjiang Scholars and Innovative Research Team in University under Grant IRT201206, and in part by the Beijing Committee of Science and Technology of China under Grant Z131110002813118. The work of Y.-W. Chen was supported by the Japanese Ministry of Education, Culture, Sports, Science and Technology through the Support Program for the Strategic Research Foundation at Private Universities 2013-2017. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Brendt Wohlberg.

C. Bai, 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, P.R. China (e-mail: baichenyan@gmail.com; jiali.gm@gmail.com; jianyu@bjtu.edu.cn).

Z. Lin is with the Key Laboratory of Machine Perception (Ministry of Education), School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, P.R. China (e-mail: zlin@pku.edu.cn). *(Corresponding author: Zhouchen Lin)*

Y.-W. Chen is with the College of Information Science and Engineering, Ritsumeikan University, Kusatsu 525-8577, Japan (e-mail: cyw2010jpcn@gmail.com).

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.2015.2409569

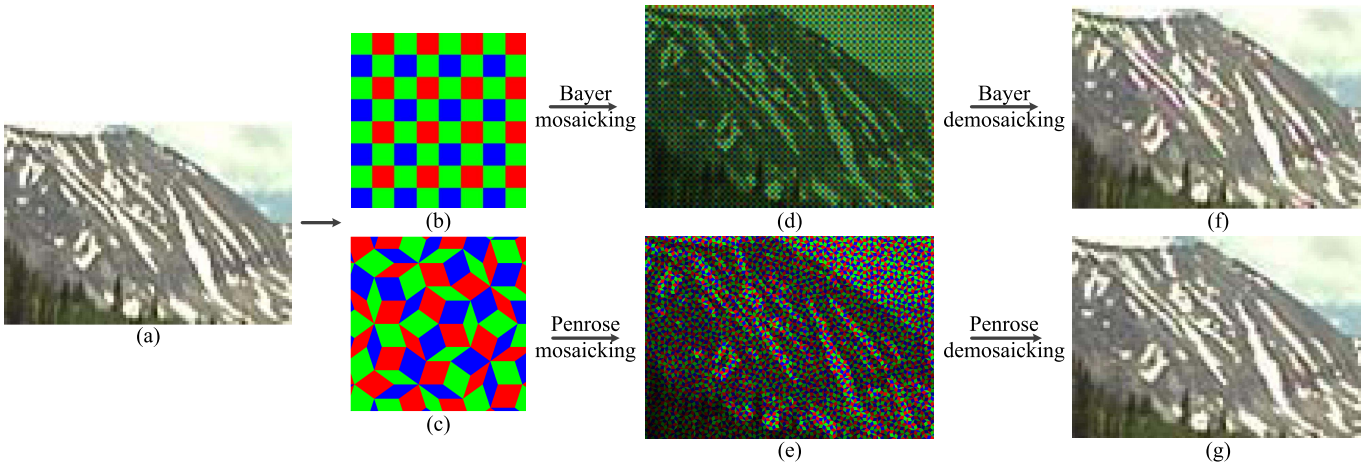


Fig. 1. Comparison of Penrose CFA and Bayer CFA. More comparisons between Penrose CFA and other regular RGB CFAs are shown in Fig. 6-8. (a) A full color image to simulate a real scene. (b) Bayer CFA. (c) Penrose CFA. (d) Bayer mosaicked image. (e) Penrose mosaicked image. (f) Bayer demosaicked image. (g) Penrose demosaicked image. **Images in this paper are best viewed on screen!**.

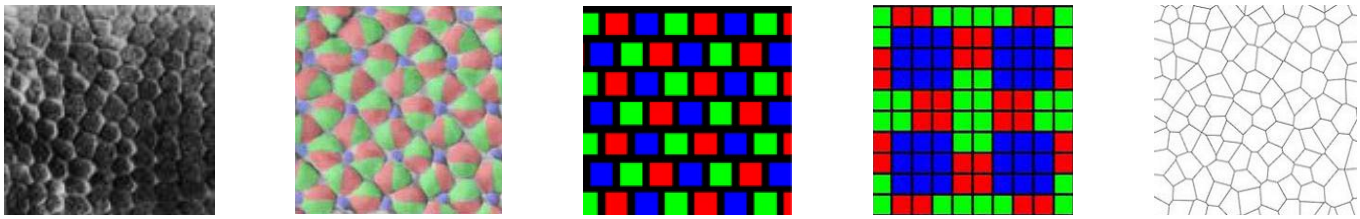


Fig. 2. Animal retina, retina-inspired CFAs, and Voronoi tessellation. (a) Human retina layout [16]. (b) Retina layout of the African cichlid fish Astatotilapia burtoni [13]. (c) Hexagonal CFA [3] designed according to human retina (a). (d) Burtoni CFA [12] designed according to the retina of Astatotilapia burtoni (b). (e) A Voronoi tessellation [14].

Our Penrose layout is also inspired by visual physiology. Some researchers have pointed out the advantages of the special structure of animal retina [3], [11] and have designed CFAs accordingly. For example, by mimicking the layout of cone cells on human retina (Fig. 2(a)), which is close to be hexagonal, Middleton and Sivaswamy proposed a near hexagonal CFA (Fig. 2(c)) [1]; and Couillaud et al. proposed the Burtoni CFA (Fig. 2(d)) [12] by mimicking the retina of African cichlid fish Astatotilapia burtoni (Fig. 2(b)) [13]. These retina-inspired CFAs have been verified to improve the image quality. However, they all simplify the irregular retina to square pixels in regular layouts. So they do not simulate the retina cell layout closely. In contrast, Penrose layout can simulate the retina layouts better (especially the retina of Astatotilapia burtoni). Although Voronoi tessellation (Fig. 2(e)) can also approximate the retina layout well [14], it is not always three-colorable[2] and is difficult for hardware manufacture due to completely different shapes of pixels.

The contributions of this paper are:

- We propose using the Penrose layout for capturing color images. Due to the uniform three-colorability of Penrose layout, the quality of color images can be guaranteed. Moreover, since Penrose layout uses only two shapes of pixels, its manufacturing is relatively easy (although not as easy as square and hexagonal layouts). It should be the first choice if an irregular layout is considered for color imaging.
- We propose a sparse representation [15] based method for Penrose demosaicking, which is more challenging than regular demosaicking. With the proposed demosaicking method, we show that Penrose layout performs better than regular layouts in terms of visual quality and S-CIELAB. This benefits from the non-square pixels and the aperiodic structure of Penrose layout which helps producing high quality full color images.

The remaining part of the paper is organized as follows. In Section II, we review the existing regular CFAs and demosaicking methods. Then we introduce our Penrose demosaicking method in Section III. In Section IV, we conduct experiments to show the effectiveness of Penrose layout. Finally, we conclude the paper and present some future work in Section V.

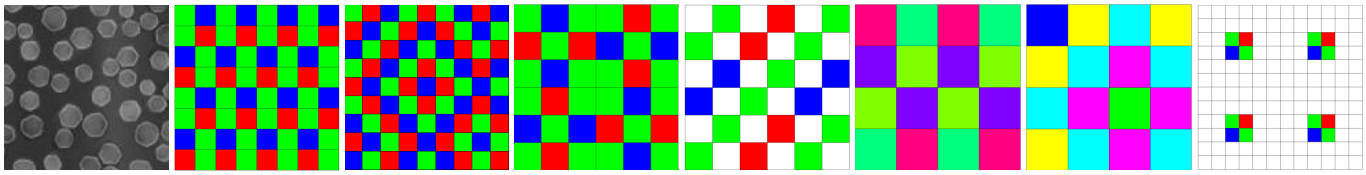
II. Related Work

To produce a color image, a digital camera needs sensors to record at least three color components, such as R, G, and B. To reduce the cost, most of the digital cameras cover the sensors with CFAs so that each pixel only records one color component. The two missing color components at each pixel have to be estimated by demosaicking methods. Although we mainly focus on CFA design, we cannot show the full color images produced by Penrose CFA without demosaicking. So in this section we review both the existing CFAs and demosaicking methods.

A. Regular CFAs

Many CFAs have been proposed under different considerations [8], [17], [18]. All of them use

[2]One may easily check that the Voronoi tessellation shown in Fig. 2(e) is not three-colorable.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 3. The particles in silver halide films and representative regular CFAs. (a) The particles in silver halide films [22]. (b) Lukac CFA [17]. (c) Random CFA [21]. (d) X-Trans CFA [22] designed according to (a). (e) Hamilton CFA [25]. (f) Hirakawa CFA [26]. (g) Hao CFA [18]. (h) Chakrabarti CFA [27].

square pixels. They differ in color components used (e.g., R, G, and B, or C, M, and Y) and component assignments (e.g., periodic or non-periodic). As images are usually displayed and stored in RGB color format, we roughly divide the existing CFAs into RGB CFAs and non-RGB CFAs.

1) RGB CFAs: The first and most commonly used RGB CFA was invented by Bayer, hence called the Bayer CFA [10] (Fig. 1(b)). It was designed based on the human visual system. As human eyes are more sensitive to green light, the sampling rate of the green component is twice as those of red and blue ones, which results in an unbalanced assignment of R, G, and B components. From a single blue noise mask pattern, three pseudo-random CFAs were proposed in [19] to eliminate the aliased spectrum. Fillfactory company also argued that the random assignments of color components are usually more robust to Moiré effect [17]. Parmar and Reeves [20] formulated an error criterion that reflects the error between the original image and the reconstructed image when viewed through the human visual system. Lukac and Plataniotis [17] provided a summary of the requirements on CFA design and the performance analysis on various RGB CFAs. They also presented the Lukac CFA (Fig. 3(b)) in [17], which is not three-colored, i.e., there exists two adjacent pixels that share a common side are assigned the same color component. Based on the blue noise characteristics of chrominance channels, Condat [21] proposed an RGB CFA with a random, uniformly three-colorable square layout (Fig. 3(c)), which yields less demosaicking visual artifacts than the Bayer CFA does. Recently, by mimicking the irregular and random arranged particles in silver halide film (Fig. 3(a)), Fujifilm company developed the Fujifilm X-Trans (X-Trans for short) CMOS sensor [22], which has been used in many Fujifilm X-series cameras. The X-Trans CFA (Fig. 3(d)) is claimed to produce less Moiré artifacts than the Bayer CFA does.

2) Non-RGB CFAs: Many non-RGB CFAs are variations of RGB CFAs. A direct substitute for R, G, and B are the subtractive colors, i.e., C, M, and Y. For example, Kodak's DCS620x used the Bayer CFA with CMY colors because CMY are more sensitive to light than RGB [8]. Some CFAs used CMYG colors to provide a compromise between light sensitivity and color quality [23]. Sony company introduced a light blue color, emerald (E), into the standard Bayer CFA to record natural images that are closer to the natural sight perception of human eyes [24]. Hamilton and Compton [25] introduced one panchromatic pixel to surround every RGB colored pixel (Fig. 3(e)).

Hirakawa and Wolfe [26] designed CFAs directly in the Fourier domain. The CFA design was posed as a problem of parameter selection in order to separate the chrominance far away from the luminance. Then each pixel in raw images by their CFAs is a linear combination of RGB components. The CFA with the best performance in their experiments is shown in Fig. 3(f). Hao et al. further formulated the design in [26] as an optimization problem to find new CFAs [18], which minimized the demosaicking error. The new CFAs were periodic tilings of 4×4 patterns and used colors other than RGB and CMY (Fig. 3(g)).

More recently, Chakrabarti et al. [27] proposed a panchromatic predominant CFA, which samples colors with the Bayer pattern very sparsely (Fig. 3(h)). They also presented a two-step approach to reconstruct full color images. The new CFA was shown to significantly reduce noise and aliasing artifacts in low-light conditions.

B. Demosaicking Methods

A huge number of image demosaicking methods have been proposed for regular CFAs [6]–[8]. Roughly speaking, demosaicking algorithms can be categorized into five categories: heuristic, directional-interpolation-based, frequency-domain-based, wavelet-based, and reconstruction-based. Almost all of them work with the Bayer CFA.

1) Heuristic Methods: Most of the heuristic methods first interpolate the G channel, then interpolate the R and B channels based on some reasonable assumptions on the inter-channel correlations. One commonly used assumption is that the hue (color ratios) within an area of an image is constant [28], [29]. The color differences can also be used to estimate the R and B channels [30]. The interpolation can also be adaptive, e.g., along edges, rather than across edges [31]–[33]. Another class of methods estimated the missing pixel values by a weighted sum of their neighboring pixels [34].

2) Directional Interpolation Methods: Directional interpolation methods first interpolate along the horizontal and vertical directions, respectively, generating two full color images. Then the two estimated values for each pixel are fused as the final recovery result. Hirakawa and Parks [35] used the local homogeneity in the CIELAB color space [36] to produce the recovery image. Kimmel [29] proposed a weighted average method to decide the edge direction, which was further improved in [37]. Paliy et al. [38] used the local polynomial approximation and the intersection of confidence intervals to fuse the directional interpolation. Menon et al. [39] presented a demosaicking method based on directional filtering and a posteriori decision. Li and Randhawa [40] first interpolated on four opposite

directions and then used a weighted median filter to produce an output image. Zhang et al. [41] first estimated local directions by gradients and then exploited non-local redundancy to improve the local color reproduction result. Kiku et al. [42] proposed the residual interpolation and incorporated it into the gradient based threshold free algorithm [43].

3) Frequency Domain Methods: The frequency domain methods use a frequency representation of the raw image, i.e., CFA mosaicked image. Alleysson et al. [44] represented a raw image as a summation of a luminance signal and two chrominance signals. Dubois [45] proposed an adaptive algorithm that handled the spectral overlap between chrominance and high-frequency luminance components. Lian et al. [46] observed two different states in the spectrum of a raw image. Based on the spectral-spatial correlation, Chung et al. [47] obtained an adaptive heterogeneity-projection with proper mask size for each pixel. In [48], Leung et al. designed a bandpass filter by the least square approach.

4) Wavelet-Based Methods: Some literatures apply the wavelet theory for demosaicking. Gunturk et al. [49] proposed an algorithm which forced similar high-frequency characteristics for the R, G, and B channels. Lu et al. [50] gave an analysis on this approach and presented a single-step algorithm. A similar method was proposed by Li in [51]. Menon and Calvagno [52] exploited a wavelet-based analysis of the luminance component to drive an adaptive interpolation algorithm.

5) Reconstruction-Based Methods: Reconstruction-based methods formulate the demosaicking process as an optimization problem that incorporates the inter-channel correlation and image priors. The maximum a posteriori probability (MAP) model [53] and the Markov random field model [54] are popular for defining the objective function. Regularization is also widely used. Keren and Osadchy [55] proposed a regularizer including a spatial smoothness and a color correlation term. Saito and Komatsu [56] used the total variation to impose smoothness on intra-channel and color differences. Menon and Calvagno [57] proposed a regularization approach to exploit global and local natural color image prior. Buades et al. [58] took into account the non-local image geometry in the method. Mairal et al. [15] introduced the sparsity prior in image demosaicking. This approach was extended in [59] with an on-line dictionary learning strategy. Moghadam et al. [60] presented a compressed sensing based demosaicking method. Recently, Condat [61] showed a joint demosaicking-deblurring application using total variation minimization.

III. Penrose Demosaicking

In this section, we present our method to demosaick the Penrose CFA. We first define the mapping between Penrose and square layouts. Then we describe our Penrose demosaicking model and algorithm. We use upper and lower case bold letters to denote matrices and vectors, respectively. Both images and patches are represented in a column vector in the order of R, G, and B.

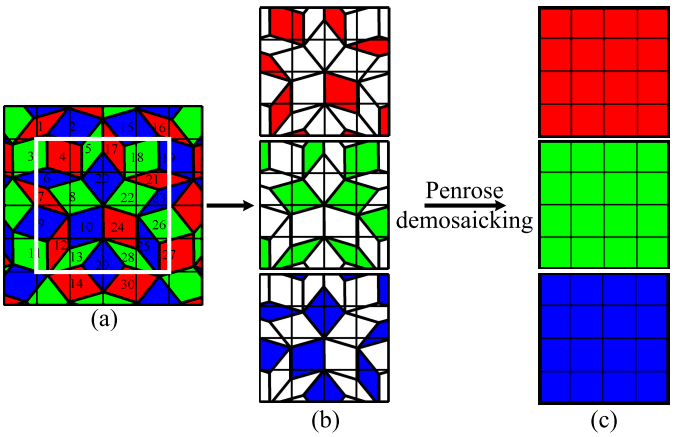


Fig. 4. Penrose demosaicking process. (a) A Penrose raw image. The horizontal and vertical thin lines indicate the square layout and the white rectangle indicates the patch of image to be demosaicked. (b) The mosaicked R, G, and B channels of the patch, respectively. (c) The demosaicked R, G, and B channels of the patch.

A. Mapping Between Penrose and Square Layouts

As in [62], we assume that the pixels have uniform photosensitivity. This implies that the contribution of a Penrose pixel to a square pixel is proportional to its area inside the square pixel and vice versa. So the mapping between the Penrose and the square layouts is as follows:

$$\mathbf{p} = \mathbf{M}\mathbf{s} + \mathbf{e}, \tag{1}$$

where $\mathbf{p}$ and $\mathbf{s}$ are the column vectors of the irradiances of all Penrose and square pixels involved, respectively, and $\mathbf{e}$ is the noise, which always exists, e.g., due to graylevel quantization.

Since color images have three channels, there are three coefficient matrices $\mathbf{M}_R$, $\mathbf{M}_G$, and $\mathbf{M}_B$, corresponding to the R, G, and B channels, respectively. So Equation (1) can be written in more detail:

$$\begin{pmatrix} \mathbf{p}_R \\ \mathbf{p}_G \\ \mathbf{p}_B \end{pmatrix} = \begin{pmatrix} \mathbf{M}_R & 0 & 0 \\ 0 & \mathbf{M}_G & 0 \\ 0 & 0 & \mathbf{M}_B \end{pmatrix} \begin{pmatrix} \mathbf{s}_R \\ \mathbf{s}_G \\ \mathbf{s}_B \end{pmatrix} + \mathbf{e}. \tag{2}$$

So $\mathbf{M}$ is a block diagonal matrix, with the coefficient matrices $\mathbf{M}_R$, $\mathbf{M}_G$, and $\mathbf{M}_B$ of the three channels being its diagonal blocks.

Take $\mathbf{M}_G$ for example, the entries of $\mathbf{M}_G$ are the ratios of the areas of the intersections between the two types of pixels to the area of square pixel. Namely,

$$\mathbf{M}_G(i, j) = \frac{\mathrm{A}(P_G(i) \cap S_G(j))}{\mathrm{A}(P_G(i))}, \tag{3}$$

where $P_G(i)$ and $S_G(j)$ are the i-th Penrose pixel and the j-th square pixel with the green component, respectively, and $\mathrm{A}(X)$ represents the area of X. We take Fig. 4 as an example. There are two green Penrose pixels, numbered with 8 and 22, that are completely inside the window outlined by a white rectangle. The corresponding $\mathbf{M}_G$ is as in Fig. 5. For instance, if one indexes the to-be-demosaicked rectangular patch column-wise, then the eighth Penrose pixel covers 0.30, 0.05, 0.51, and 0.14 of the second, third, sixth, and seventh square pixels, respectively.

$$\mathbf{M}_G = \begin{pmatrix} 0.00 & 0.30 & 0.05 & 0.00 & 0.00 & 0.51 & 0.14 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.50 & 0.15 & 0.00 & 0.00 & 0.30 & 0.05 & 0.00 \end{pmatrix}$$

Fig. 5. An illustration of the coefficient matrix that maps a square layout with the green component into a Penrose one.

B. Penrose Demosaicking Model

We can see from Fig. 1 and 4 that for a Penrose raw image none of the pixel values is equal to those of the demosaicked image. In contrast, for a square layout raw image, the pixel values of the raw image will be maintained after demosaicking. So Penrose demosaicking is inherently more challenging than regular demosaicking. Although there are universal demosaicking methods [57], [63], [64] appropriate for the Bayer CFA and various non-Bayer CFAs, they are limited to square layouts and cannot be directly used for Penrose demosaicking.

Natural image statistics suggests that image patches can be approximated by a sparse linear combination of elements from a well-chosen over-complete dictionary [65]. Motivated by this assumption, Mairal et al. [15] proposed a sparse representation based demosaicking model for regular pixel layout which viewed the Bayer raw image as a noisy image with some pixel values unknown. So the demosaicking problem was modeled as a denoising process. The demosaicking model for the square layout is as follows:

$$\min_{\mathbf{s},\boldsymbol{\alpha}} \frac{1}{2}\|\mathcal{M}(\mathbf{s}-\mathbf{D}\boldsymbol{\alpha})\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_0 + \frac{\mu}{2}\|\mathcal{M}(\mathbf{s})-\mathbf{r}\|_2^2, \quad (4)$$

where $\mathbf{D}$ is a dictionary learned from clean full color patches, $\boldsymbol{\alpha}$ is the sparse code, $\|\boldsymbol{\alpha}\|_0$ denotes the number of nonzero elements of vector $\boldsymbol{\alpha}$ which is a non-convex function w.r.t. $\boldsymbol{\alpha}$, $\|\cdot\|_2$ is the l_2 norm, $\mathbf{r}$ is a patch of Bayer raw image, and $\mathcal{M}$ is the mosaicking operator, i.e., the linear mapping that only keeps the values of the three color components at their sampled pixels. The demosaicked patch is the optimal $\mathbf{s}$ of problem (4).

Inspired by sparse representation [65] and Mairal et al. [15], we propose the following Penrose demosaicking model:

$$\min_{\mathbf{s},\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{s}-\mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 + \frac{\mu}{2}\|\mathbf{M}\mathbf{s}-\mathbf{p}\|_2^2, \quad (5)$$

where $\mathbf{p}$ and $\mathbf{s}$ are vectors collecting Penrose and square pixel values, respectively, $\|\cdot\|_1$ is the l_1 norm which is the convex relaxation of l_0 pseudo norm [66], and $\mathbf{M}$ is the coefficient matrix that maps square pixels into Penrose ones. The fidelity term $\frac{1}{2}\|\mathbf{M}\mathbf{s}-\mathbf{p}\|_2^2$ results from Equation (1). The terms $\|\mathbf{s}-\mathbf{D}\boldsymbol{\alpha}\|_2^2+\lambda\|\boldsymbol{\alpha}\|_1$ are the image prior to encourage the patch in square pixels to be represented by a few number of atoms in the dictionary $\mathbf{D}$. The parameter λ balances sparsity of $\boldsymbol{\alpha}$ and fidelity of the approximation to $\mathbf{s}$, while μ balances image prior and fidelity of the mosaicking process.

However, Penrose pixels cannot fill in a rectangular region exactly. So there must exist incomplete Penrose pixels in the to-be-demosaicked rectangular patch, e.g., the fifth Penrose pixel in Fig. 4 successively covers 0.24, 0.74, and 0.02 of three square pixels, which are vertically adjacent and only the last two are inside the white rectangle. To account for this effect, we modify (5) slightly as follows:

$$\min_{\mathbf{s},\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{s}-\mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 + \frac{\mu}{2}\|\mathbf{M}\mathbf{s}-\mathbf{W}\mathbf{p}\|_2^2, \quad (6)$$

where $\mathbf{W}=\mathrm{diag}(\mathbf{w})$ and $\mathbf{w}$ records the ratios of the areas of Penrose pixels inside the patch to the areas of Penrose pixels. For example, in Fig. 4, let the fifth and eighth green Penrose pixels correspond to the i-th and j-th entries of $\mathbf{p}$, respectively. There is 0.76 of the area of the fifth Penrose pixel inside the patch. So the nonzero entries of the i-th row of $\mathbf{M}$ are 0.74 and 0.02, and $\mathbf{w}(i) = 0.76$. The eighth Penrose pixel is completely inside the patch. So the nonzero entries of the j-th row of $\mathbf{M}$ are as those of the first row of $\mathbf{M}_G$ in Fig. 5, and $\mathbf{w}(j)=1$. Introducing $\mathbf{W}$ is consistent with the uniform photosensitivity assumption of sensors [62].

Let $\mathbf{x} = (\boldsymbol{\alpha}^T, \mathbf{s}^T)^T$ in problem (6), our Penrose demosaicking model is finally formulated as:

$$\min_{\mathbf{x}} \frac{1}{2}\|\,(-\mathbf{D}\ \mathbf{I})\,\mathbf{x}\|_2^2 + \lambda\|\,(\mathbf{I}\ \mathbf{0})\,\mathbf{x}\|_1 + \frac{\mu}{2}\|\,(\mathbf{0}\ \mathbf{M})\,\mathbf{x}-\mathbf{W}\mathbf{p}\|_2^2, \quad (7)$$

where $\mathbf{I}$ and $\mathbf{0}$ are the identity and zero matrices, respectively. It is a convex problem w.r.t. $\mathbf{x}$.

C. Alternating Direction Method (ADM)

The optimization problem (7) can be solved by the alternating direction method (ADM). The convergence of ADM for solving convex problems with two blocks is well established [67]. ADM updates the blocks alternately by minimizing the augmented Lagrangian function. For the following model problem:

$$\min_{\mathbf{x},\mathbf{y}} \ f(\mathbf{x})+g(\mathbf{y}), \quad \text{s.t. } \mathcal{A}(\mathbf{x})+\mathcal{B}(\mathbf{y})=\mathbf{c}, \quad (8)$$

where $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{c}$ can be either vectors or matrices, f and g are convex functions, and $\mathcal{A}$ and $\mathcal{B}$ are linear mappings, ADM operates on the following augmented Lagrangian function:

$$\begin{aligned}\mathcal{L}(\mathbf{x},\mathbf{y},\mathbf{z}) = {} & f(\mathbf{x})+g(\mathbf{y})+\langle \mathbf{z}, \mathcal{A}(\mathbf{x})+\mathcal{B}(\mathbf{y})-\mathbf{c}\rangle \\ & + \frac{\gamma}{2}\|\mathcal{A}(\mathbf{x})+\mathcal{B}(\mathbf{y})-\mathbf{c}\|_2^2, \end{aligned} \quad (9)$$

where $\mathbf{z}$ is the Lagrange multiplier, $\langle\cdot,\cdot\rangle$ is the inner product, and $\gamma > 0$ is the penalty parameter. ADM updates $\mathbf{x}$ and $\mathbf{y}$ alternately by minimizing $\mathcal{L}$ w.r.t. $\mathbf{x}$ and $\mathbf{y}$, respectively. The iterations go as follows:

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x},\mathbf{y}_k,\mathbf{z}_k), \quad (10)$$

$$\mathbf{y}_{k+1} = \arg\min_{\mathbf{y}} \mathcal{L}(\mathbf{x}_{k+1},\mathbf{y},\mathbf{z}_k), \quad (11)$$

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \gamma\,(\mathcal{A}(\mathbf{x}_{k+1})+\mathcal{B}(\mathbf{y}_{k+1})-\mathbf{c}). \quad (12)$$

ADM is advantageous when subproblems (10)-(11) are easily solvable, e.g., having closed form solutions. For more details about ADM, we refer the readers to [67].

D. Solving the Penrose Demosaicking Model by ADM

When applying ADM to (7), we first introduce an auxiliary variable $\mathbf{y}$ and reformulate it as:

$$\min_{\mathbf{x},\mathbf{y}} \frac{1}{2}\|(-\mathbf{D}\ \mathbf{I})\,\mathbf{x}\|_2^2 + \lambda\|\mathbf{y}\|_1 + \frac{\mu}{2}\|(\mathbf{0}\ \mathbf{M})\,\mathbf{x} - \mathbf{W}\mathbf{p}\|_2^2, \quad \text{s.t.}\ \ \mathbf{y} = (\mathbf{I}\ \mathbf{0})\,\mathbf{x}. \tag{13}$$

The augmented Lagrangian function of problem (13) is:

$$\mathcal{L}(\mathbf{x},\mathbf{y},\mathbf{z}) = \frac{1}{2}\|(-\mathbf{D}\ \mathbf{I})\,\mathbf{x}\|_2^2 + \lambda\|\mathbf{y}\|_1 + \frac{\mu}{2}\|(\mathbf{0}\ \mathbf{M})\,\mathbf{x} - \mathbf{W}\mathbf{p}\|_2^2 + \langle \mathbf{z}, \mathbf{y} - (\mathbf{I}\ \mathbf{0})\,\mathbf{x}\rangle + \frac{\gamma}{2}\|\mathbf{y} - (\mathbf{I}\ \mathbf{0})\,\mathbf{x}\|_2^2. \tag{14}$$

Then by ADM problem (13) can be solved via the following iterations:

$$\begin{aligned}
\mathbf{x}_{k+1} &= \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y}_k, \mathbf{z}_k) \\
&= \arg\min_{\mathbf{x}} \frac{1}{2}\|(-\mathbf{D}\ \mathbf{I})\,\mathbf{x}\|_2^2 + \frac{\mu}{2}\|(\mathbf{0}\ \mathbf{M})\,\mathbf{x} - \mathbf{W}\mathbf{p}\|_2^2 + \frac{\gamma}{2}\|\mathbf{y}_k - (\mathbf{I}\ \mathbf{0})\,\mathbf{x} + \mathbf{z}_k/\gamma\,\|_2^2 \\
&= \mathbf{H}^{-1}\begin{pmatrix}\gamma\,\mathbf{y}_k + \mathbf{z}_k \\ \mu\mathbf{M}^T\mathbf{W}\mathbf{p}\end{pmatrix}, \qquad (15)\\
\mathbf{y}_{k+1} &= \arg\min_{\mathbf{y}} \mathcal{L}(\mathbf{x}_{k+1}, \mathbf{y}, \mathbf{z}_k) \\
&= \arg\min_{\mathbf{y}} \lambda\|\mathbf{y}\|_1 + \frac{\gamma}{2}\|\mathbf{y} - (\mathbf{I}\ \mathbf{0})\,\mathbf{x}_{k+1} + \mathbf{z}_k/\gamma\,\|_2^2 \qquad (16)\\
&= \mathcal{S}_{\lambda/\gamma}\left((\mathbf{I}\ \mathbf{0})\,\mathbf{x}_{k+1} - \mathbf{z}_k/\gamma\right),\\
\mathbf{z}_{k+1} &= \mathbf{z}_k + \gamma\,(\mathbf{y}_{k+1} - (\mathbf{I}\ \mathbf{0})\,\mathbf{x}_{k+1}), \qquad (17)
\end{aligned}$$

where $\mathbf{H} = \begin{pmatrix}\mathbf{D}^T\mathbf{D} + \gamma\,\mathbf{I} & -\mathbf{D}^T \\ -\mathbf{D} & \mathbf{I} + \mu\mathbf{M}^T\mathbf{M}\end{pmatrix}$ and $\mathcal{S}_\varepsilon(x) = \mathrm{sgn}(x)\max(|x| - \varepsilon, 0)$ is the soft thresholding operator [68].

To demosaick a large image, we may demosaick it patch by patch, where nearby patches overlap by several pixels. Then we average the values of overlapping pixels to get the demosaicked image. In this way, not only the inverse of huge matrix $\mathbf{H}$ is avoided, part of the components in $\mathbf{H}^{-1}$ can be reused when demosaicking different patches. Actually, by the result in [69],

$$\mathbf{H}^{-1} = \begin{pmatrix}(\mathbf{D}^T\mathbf{D} + \gamma\,\mathbf{I})^{-1} + \mathbf{C}\mathbf{S}^{-1}\mathbf{C}^T & \mathbf{C}\mathbf{S}^{-1} \\ \mathbf{S}^{-1}\mathbf{C}^T & \mathbf{S}^{-1}\end{pmatrix},$$

where $\mathbf{S} = \mathbf{I} + \mu\mathbf{M}^T\mathbf{M} - \mathbf{D}\mathbf{C}$ is the Schur complement of $\mathbf{D}^T\mathbf{D} + \gamma\,\mathbf{I}$ in $\mathbf{H}$ and $\mathbf{C} = (\mathbf{D}^T\mathbf{D} + \gamma\,\mathbf{I})^{-1}\mathbf{D}^T$. So we only have to recompute $\mathbf{S}^{-1}$ for different patches, where $\mathbf{M}$ varies. We can further dismiss the storage of $\mathbf{H}^{-1}$ and speed up the computation by decomposing $\mathbf{x}$ into two components:

$$\begin{aligned}
\mathbf{s}_{k+1} = (\mathbf{0}\ \mathbf{I})\,\mathbf{x}_{k+1} &= \begin{pmatrix}\mathbf{S}^{-1}\mathbf{C}^T & \mathbf{S}^{-1}\end{pmatrix}\begin{pmatrix}\gamma\,\mathbf{y}_k + \mathbf{z}_k \\ \mu\mathbf{M}^T\mathbf{W}\mathbf{p}\end{pmatrix} \\
&= \mathbf{S}^{-1}\left(\mathbf{C}^T(\gamma\,\mathbf{y}_k + \mathbf{z}_k) + \mu\mathbf{M}^T\mathbf{W}\mathbf{p}\right), \qquad (18)
\end{aligned}$$

and

$$\begin{aligned}
\boldsymbol{\alpha}_{k+1} &= (\mathbf{I}\ \mathbf{0})\,\mathbf{x}_{k+1} \\
&= \begin{pmatrix}(\mathbf{D}^T\mathbf{D} + \gamma\,\mathbf{I})^{-1} + \mathbf{C}\mathbf{S}^{-1}\mathbf{C}^T & \mathbf{C}\mathbf{S}^{-1}\end{pmatrix}\begin{pmatrix}\gamma\,\mathbf{y}_k + \mathbf{z}_k \\ \mu\mathbf{M}^T\mathbf{W}\mathbf{p}\end{pmatrix} \\
&= (\mathbf{D}^T\mathbf{D} + \gamma\,\mathbf{I})^{-1}(\gamma\,\mathbf{y}_k + \mathbf{z}_k) + \mathbf{C}\mathbf{s}_{k+1}. \qquad (19)
\end{aligned}$$

Algorithm 1 Algorithm for Solving Problem (13)

Input: Vector $\mathbf{p}$ of values of Penrose pixels intersecting a to-be-demosaicked patch, pre-learned dictionary $\mathbf{D}$, positive parameters λ, μ, $\gamma = 1$, and $\varepsilon = 0.1$, $\mathbf{y}_0 = \mathbf{0}$, $\mathbf{z}_0 = \mathbf{0}$.

1: **while** the stopping conditions (22)-(23) are not met **do**
2: Update $\mathbf{s}$ by (18).
3: Update $\boldsymbol{\alpha}$ by (19).
4: Update $\mathbf{y}$ by (20).
5: Update the multiplier $\mathbf{z}$ by (21).
6: **end while**

Output: Demosaicked patch $\mathbf{s}$ in square layout.

Then we update $\mathbf{y}$ and $\mathbf{z}$ as

$$\mathbf{y}_{k+1} = \mathcal{S}_{\lambda/\gamma}\,(\boldsymbol{\alpha}_{k+1} - \mathbf{z}_k/\gamma\,), \tag{20}$$
$$\mathbf{z}_{k+1} = \mathbf{z}_k + \gamma\,(\mathbf{y}_{k+1} - \boldsymbol{\alpha}_{k+1}). \tag{21}$$

The stopping criteria are:

$$\|\mathbf{y}_{k+1} - \boldsymbol{\alpha}_{k+1}\|_\infty < \varepsilon, \ \text{and} \tag{22}$$
$$\max\{\|\mathbf{y}_{k+1} - \mathbf{y}_k\|_\infty, \|\boldsymbol{\alpha}_{k+1} - \boldsymbol{\alpha}_k\|_\infty, \|\mathbf{s}_{k+1} - \mathbf{s}_k\|_\infty\} < \varepsilon, \tag{23}$$

which account for the primal residual and the dual residual, respectively [67].

We summarize the whole solution process in Algorithm 1.

IV. Experiments

In this section, we first test Penrose layout on benchmark images and compare it with regular RGB CFAs with the proposed demosaicking method, both quantitatively and visually. The four chosen regular RGB CFAs include: the most commonly used Bayer CFA [10], which is not uniformly colored (Fig. 1 (b)), the Lukac CFA [17], which is not three-colored (Fig. 3(b)), the X-Trans CFA [22], which simulates the irregularity and random arrangement of pixels in square layout (Fig. 3(d)), and the Random CFA [21], which is random (and hence also non-periodic) and is uniformly three-colored in square layout (Fig. 3(c)). A difference between Penrose CFA and Random CFA is that the latter uses square pixels. We do not compare with non-RGB CFAs because the Penrose CFA uses R, G, and B colors.

When we demosaick from the raw images by regular RGB CFAs, the Penrose demosaicking model (see (6)) can be simplified, which we call square soft demosaicking. Its demosaicking model is:

$$\min_{\mathbf{s},\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{s} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 + \frac{\mu}{2}\|\mathcal{M}(\mathbf{s}) - \mathbf{r}\|_2^2, \tag{24}$$

where the notations are the same as those described in subsection III-B.

For reference, we also present the results of the state-of-the-art demosaicking methods on the Bayer CFA.

A. Experiment Settings

1) Simulation of Penrose Raw Images: Since the Penrose CFA has not been manufactured, we need to generate a three-colored rhombus Penrose tiling to simulate it. In the literature, several methods have been developed for generating Penrose tilings [4], [70]–[73]. Also, we can color

a rhombus Penrose tiling with the coloring algorithm proposed by Sibley and Wagon [9]. Since the pixel shapes of Penrose and square layouts are different, as in [5] the two layouts are considered to have the same resolution if they have the same number of pixels within a given area. So we scale the size of Penrose layout so that the two layouts have the same number of pixels in the imaging area. Then for each patch, we compute the coefficient matrix $\mathbf{M}$ of the mapping between the two layouts. We regard every rhombus pixel and square pixel as polygonal regions, whose vertex coordinates are recorded during the generation of the two layouts. For example, when we compute the coefficient matrix of the green channel $\mathbf{M}_G$ by (3), the areas of rhombus pixels and square pixels in $\mathbf{M}_G$ (see (3)) are estimated as that of the two polygonal regions, and so is the area of the intersection between them.

We scale Penrose layout to match the size of square layout as follows. Let the area of square pixel be 1 and the size of square layout be $u \times v$ pixels, then we have the following four equations with four unknowns:

$$\begin{cases} a_1/a_2 = \phi, \\ n_1/n_2 = \phi, \\ n_1 + n_2 = uv, \\ n_1 a_1 + n_2 a_2 = uv, \end{cases} \tag{25}$$

where a_1 and a_2 are the areas of thick and thin rhombus pixels, respectively, n_1 and n_2 are their numbers, respectively, and $\phi = \frac{1+\sqrt{5}}{2}$ is the golden number. (25) has a unique solution. If the areas of thick and thin rhombus pixels in the initial Penrose tiling are $\hat{a}_1$ and $\hat{a}_2$, respectively, then the scaling factor can be computed as $\sqrt{a_1/\hat{a}_1}$ (or equivalently $\sqrt{a_2/\hat{a}_2}$). We next scale the initial Penrose tiling with the scaling factor and crop a $u \times v$ rectangular region from the scaled Penrose tiling as Penrose layout.

We simulate the imaging process as follows (Fig. 1). We first choose a full color image, which is also the ground truth, to simulate the scene. Then we obtain the raw image patches in both Penrose layout and square layout according to the mosaicking process shown in Equation (2). For instance, the green pixel values of the Penrose raw patch are computed by $\mathbf{p}_G = \mathbf{M}_G \mathbf{s}_G$, where the notations are the same as those described in subsection III-A. However, the pixel values in $\mathbf{p}_G$ are real numbers. So we also quantize $\mathbf{p}_G$ using MATLAB's default `uint8` function to generate the final 8-bit grayscale Penrose raw patch. As there is always graylevel quantization, we can only use an inexact data-fitting term in the Penrose demosaicking model (see (6)). The imaging simulation for square layouts with RGB CFAs can be simplified as a sub-sampling process, i.e., keeping only one color for each pixel as specified by each RGB CFA.

2) Dataset and Evaluation Metrics: We test on the most commonly used Kodak PhotoCD dataset [74], which contains 24 images that are film captured. They are digitized and then padded to a resolution of 512×768 pixels and encoded in RGB with 8 bits per channel.

We directly use the dictionary $\mathbf{D}$ learned for SC [15], which is publicly available.[3] It is also the initial dictionary for LSSC [59]. The pre-learned dictionary has 256 atoms, each of dimension $8 \times 8 \times 3$ for 8×8 RGB patches. We demosaick patch by patch, where the patch size is 8×8 pixels and the overlap between nearby patches is seven pixels. Note that the dictionary $\mathbf{D}$ for SC [15] is learned using the online procedure of [75], which is detailed in [59]. In contrast, the original dictionary for SC [15] is learned by K-SVD [76], which is a batch learning method and unusable with large-scale training data [59].

We select the first five images from the Kodak dataset for parameter tuning. It results in $\lambda = 1$ and $\mu = 8$ for Bayer CFA [10], $\lambda = 0.5$ and $\mu = 8$ for Lukac CFA [17], $\lambda = 0.8$ and $\mu = 8$ for X-Trans CFA [22], $\lambda = 0.5$ and $\mu = 8$ for Random CFA [21], and $\lambda = 0.35$ and $\mu = 3$ for Penrose CFA. They are the parameters that give the best average CPSNR values on the selected images. Following SC [15] and LSSC [59], these parameters are fixed when testing on all 24 images in the Kodak dataset. As the Penrose CFA has incomplete pixels at the boundary, we also exclude an 8-pixel border (one patch width) to eliminate the boundary effects. We implement Penrose demosaicking and square soft demosaicking in MATLAB. With our unoptimized codes, Penrose demosaicking takes about 49.7 minutes to demosaick an image of size 512×768 pixels on a 4.00 GHz Quad Core machine.

We adopt two objective image quality measures, CPSNR and S-CIELAB $\Delta E^{\star}_{ab}$ [36], to evaluate the performance of all CFAs in comparison. CPSNR is based on computing the pixel-wise color difference between the demosaicked image and the ground truth image. It is not very related to visual perception. In contrast, the S-CIELAB $\Delta E^{\star}_{ab}$ metric uses a perceptual model to measure the spatial color reproduction accuracy of a recovered image as seen by a human. A larger $\Delta E^{\star}_{ab}$ value may indicate larger error and poorer visual quality.

However, neither CPSNR nor S-CIELAB complies with human vision perfectly. Namely, images with better CPSNR or S-CIELAB values are not necessarily of higher visual quality [77], [78]. There are often subtle artifacts in the demosaicked images, such as false color and zipper effect [34], [35], which can degrade the visual quality significantly but may only affect the values of CPSNR and S-CIELAB very slightly. So we also compare the demosaicking results by visual evaluation, which we deem more important than quantitative evaluation.

B. Comparison With Regular RGB CFAs

We first quantitatively compare the Penrose CFA with regular RGB CFAs. The upper left part of Table I gives the individual and average CPSNR values of each CFA on the benchmark test images, while the lower left part of Table I gives the S-CIELAB values. The best values in the left part of Table I are in boldface.

In terms of CPSNR, we can see that Random CFA [21] outperforms the other CFAs on both individual image and the whole dataset. Lukac CFA [17] has competitive CPSNR values. With the proposed demosaicking method, the Penrose CFA is inferior in terms of CPSNR on the test images. One key reason is that in the Penrose CFA *none of the color components of the reconstructed regular color images is equal*

[3] http://lear.inrialpes.fr/people/mairal/software.php.

TABLE I

EVALUATION ON THE KODAK PHOTOCD DATASET. THE LEFT PART SHOWS THE COMPARISON BETWEEN THE PENROSE CFA AND FOUR REGULAR RGB CFAS. THE RIGHT PART SHOWS THE RESULTS OF THE STATE-OF-THE-ART DEMOSAICKING METHODS ON THE BAYER CFA, PRESENTED JUST FOR REFERENCE. "EVAL.", "DIR. INTERP.", "HEU.", AND "FREQ." STAND FOR "EVALUATION", "DIRECTIONAL INTERPOLATION", "HEURISTIC", AND "FREQUENCY", RESPECTIVELY. THE INDIVIDUAL AND AVERAGE CPSNR AND S-CIELAB $\Delta E^{\star}_{ab}$ VALUES ARE REPORTED

		Regular RGB CFAs					Dir. Interp.		Heu.	Freq.	Wavelet		Reconstruction			
Eval. Metric	No.	Bayer [10]	Lukac [17]	X-Trans [22]	Random [21]	Penrose	LPA [38]	RI [42]	Lu [34]	LSLCD [48]	WL [52]	AP1 [50]	SC [15]	LSSC [59]	RAD [57]	CD8 [60]
CPSNR	01	38.97	39.43	38.60	**39.77**	36.66	40.46	39.06	36.15	39.51	38.59	37.99	40.85	41.36	38.30	37.81
	02	39.89	41.08	40.29	**41.49**	40.28	41.32	40.06	39.48	41.55	40.39	39.50	41.75	42.21	39.85	38.61
	03	41.23	41.63	40.91	**41.66**	40.76	43.50	41.71	41.09	43.26	42.03	41.55	43.20	44.25	42.15	37.28
	04	40.27	40.20	39.40	**40.89**	40.12	40.89	40.35	39.88	41.19	40.16	40.07	42.24	42.48	40.81	41.05
	05	37.53	36.85	36.61	**37.57**	35.79	37.56	37.80	36.96	38.76	37.76	37.54	38.79	39.52	38.04	37.91
	06	39.49	40.43	39.73	**40.80**	37.86	40.95	40.90	36.87	40.98	40.09	38.82	41.34	41.74	39.88	39.34
	07	41.41	41.46	40.83	**41.91**	40.98	43.03	41.99	41.40	43.57	41.74	41.80	43.26	44.03	42.45	41.59
	08	35.36	36.85	36.30	**37.63**	34.49	37.14	36.59	34.26	36.09	36.37	35.49	37.47	37.62	36.05	35.49
	09	41.17	41.71	41.49	**42.43**	40.79	43.52	42.56	40.78	42.94	42.47	41.92	43.21	43.84	42.13	42.40
	10	41.24	41.72	41.44	**42.29**	40.85	42.67	42.47	40.87	42.82	42.43	42.07	42.89	43.21	42.41	42.27
	11	38.95	40.11	39.14	**40.44**	38.29	40.55	40.28	38.09	40.46	39.70	39.39	41.25	41.55	39.75	39.22
	12	42.14	42.56	42.23	**43.06**	41.87	44.00	43.30	41.03	43.71	43.39	42.70	44.28	44.86	43.22	43.49
	13	35.08	**35.17**	34.24	35.11	32.70	36.11	35.43	32.46	36.56	34.55	34.51	36.21	36.40	34.87	34.19
	14	36.29	36.65	36.19	**36.95**	35.66	37.01	36.34	36.84	37.87	35.96	35.85	37.78	38.89	36.17	36.27
	15	39.12	39.38	38.87	**39.96**	39.09	40.09	38.85	38.53	40.69	39.35	39.30	41.03	41.73	39.89	39.30
	16	42.40	43.57	42.87	**44.01**	41.01	44.01	44.30	39.65	44.37	43.81	42.13	44.36	44.89	43.28	42.65
	17	40.66	40.82	40.62	**41.36**	39.69	41.83	41.79	39.80	41.97	41.13	41.48	41.81	42.06	41.53	41.15
	18	36.91	**37.17**	36.51	**37.17**	35.76	37.36	37.52	36.03	37.83	36.94	37.32	38.13	38.17	37.40	37.05
	19	38.81	40.68	40.20	**41.38**	38.54	41.51	41.31	38.74	41.02	40.89	40.02	41.86	42.28	40.04	40.15
	20	40.33	40.74	40.24	**41.32**	39.57	41.43	41.23	40.06	41.41	40.69	40.70	41.92	42.23	41.00	40.72
	21	38.55	39.84	39.20	**40.13**	37.77	39.65	39.67	37.35	39.65	38.96	38.88	40.60	40.70	39.27	38.48
	22	37.55	38.01	37.45	**38.25**	37.23	38.52	38.50	37.78	38.92	38.14	37.77	38.81	39.33	38.34	38.40
	23	40.98	41.77	41.02	**42.26**	41.74	43.92	42.02	41.54	44.28	42.08	41.92	43.47	44.34	42.38	38.75
	24	35.29	**35.36**	34.91	**35.36**	33.97	35.35	35.25	33.99	35.70	34.65	34.82	35.53	35.82	35.55	35.37
	Avg.	39.15	39.72	39.14	**40.13**	38.40	40.52	39.97	38.32	40.63	39.68	39.31	40.92	41.40	39.78	39.12
S-CIELAB	01	1.14	1.13	1.37	1.19	**0.83**	0.93	0.98	1.43	1.06	1.09	1.25	0.89	0.80	1.19	1.17
	02	0.97	0.74	0.89	0.64	**0.57**	0.62	0.75	1.08	0.58	0.71	0.78	0.59	0.55	0.72	1.05
	03	0.92	0.72	0.87	0.73	**0.56**	0.45	0.49	0.88	0.48	0.53	0.56	0.47	0.41	0.55	0.80
	04	0.96	0.88	1.19	0.80	**0.58**	0.72	0.70	0.94	0.65	0.75	0.74	0.57	0.60	0.72	0.75
	05	1.23	1.41	1.54	1.39	**1.04**	1.29	1.03	1.23	0.94	1.10	1.14	0.93	0.82	1.11	1.01
	06	1.03	0.83	1.02	0.97	**0.66**	0.76	0.72	1.20	0.73	0.77	0.94	0.70	0.64	0.85	0.82
	07	0.73	0.77	0.94	0.83	**0.57**	0.52	0.56	0.85	0.48	0.65	0.64	0.50	0.45	0.58	0.58
	08	1.71	1.40	1.82	1.40	**1.09**	1.23	1.26	1.60	1.44	1.32	1.52	1.16	1.11	1.40	1.43
	09	0.95	0.82	0.95	0.76	**0.55**	0.54	0.55	0.85	0.57	0.58	0.65	0.55	0.53	0.64	0.65
	10	0.89	0.74	0.81	0.73	**0.52**	0.56	0.55	0.83	0.54	0.55	0.60	0.53	0.52	0.61	0.59
	11	1.14	0.87	1.14	0.92	**0.63**	0.72	0.68	1.00	0.72	0.75	0.83	0.66	0.63	0.79	0.83
	12	0.68	0.60	0.64	0.72	**0.44**	0.44	0.47	0.78	0.47	0.49	0.53	0.43	0.40	0.52	0.46
	13	1.77	1.71	2.15	1.90	**1.30**	1.55	1.45	1.92	1.42	1.57	1.73	1.36	1.32	1.62	1.67
	14	1.25	1.12	1.35	1.17	**0.92**	0.97	0.95	1.17	0.91	1.12	1.13	0.86	0.76	1.07	1.13
	15	1.01	0.84	0.97	0.84	**0.57**	0.64	0.71	0.89	0.58	0.73	0.71	0.56	0.52	0.68	0.74
	16	0.76	0.62	0.78	0.63	**0.50**	0.53	0.49	0.96	0.50	0.54	0.66	0.50	0.46	0.59	0.58
	17	0.73	0.70	0.86	0.64	**0.47**	0.54	0.50	0.75	0.50	0.56	0.56	0.51	0.51	0.55	0.57
	18	1.41	1.25	1.50	1.32	**0.93**	1.10	0.99	1.32	0.99	1.08	1.06	0.94	0.94	1.04	1.09
	19	1.27	0.95	1.17	0.89	**0.68**	0.71	0.70	1.06	0.74	0.75	0.83	0.68	0.66	0.84	0.84
	20	0.75	0.72	0.93	0.68	**0.52**	0.54	0.68	0.80	0.53	0.66	0.66	0.49	0.47	0.59	0.56
	21	1.41	0.98	1.20	0.97	**0.70**	0.88	0.84	1.16	0.87	0.92	0.97	0.78	0.77	0.94	0.96
	22	1.37	1.24	1.58	1.26	**0.88**	0.98	0.97	1.20	0.88	1.02	1.05	0.89	0.83	0.97	0.96
	23	0.83	0.75	0.97	0.77	**0.52**	0.46	0.53	0.84	0.44	0.58	0.56	0.46	0.43	0.53	0.75
	24	1.22	1.19	1.51	1.23	**0.88**	1.14	1.11	1.32	1.00	1.07	1.14	0.98	0.96	1.06	1.00
	Avg.	1.09	0.96	1.17	0.97	**0.70**	0.78	0.78	1.09	0.75	0.83	0.88	0.71	0.67	0.84	0.87

to those of the ground truth. In contrast, for regular RGB CFAs *one third of the pixel color values are identical to the ground truth*. So regular CFAs are inherently advantageous in CPSNR. On the other hand, although the 1.73dB gap between Random CFA and Penrose CFA seems a bit large, the difference in average per-pixel error is only 0.55 graylevel, which could be compensated if smart quantization is adopted.[4]

It is well known that CPSNR is not a good measure for visual quality [77]. So people often use S-CIELAB values, which is closer to human perception than CPSNR, for evaluation as well. In terms of S-CIELAB, we can see that the Penrose CFA outperforms regular RGB CFAs significantly on both individual image and the whole dataset.

As the visual quality is typically the most important evaluation for image demosaicking, we present part of the visual comparison[5] in Fig. 6-8. We can see that the visual quality of Penrose CFA is better than that of regular RGB CFAs,

[4]In our experiments, we simply used Matlab's default `uint8` function to quantize Penrose demosaicked images.

[5]All the 24 demosaicked images of each method on the Kodak dataset can be found at: http://www.cis.pku.edu.cn/faculty/vision/zlin/PenroseDemosaicking.htm.

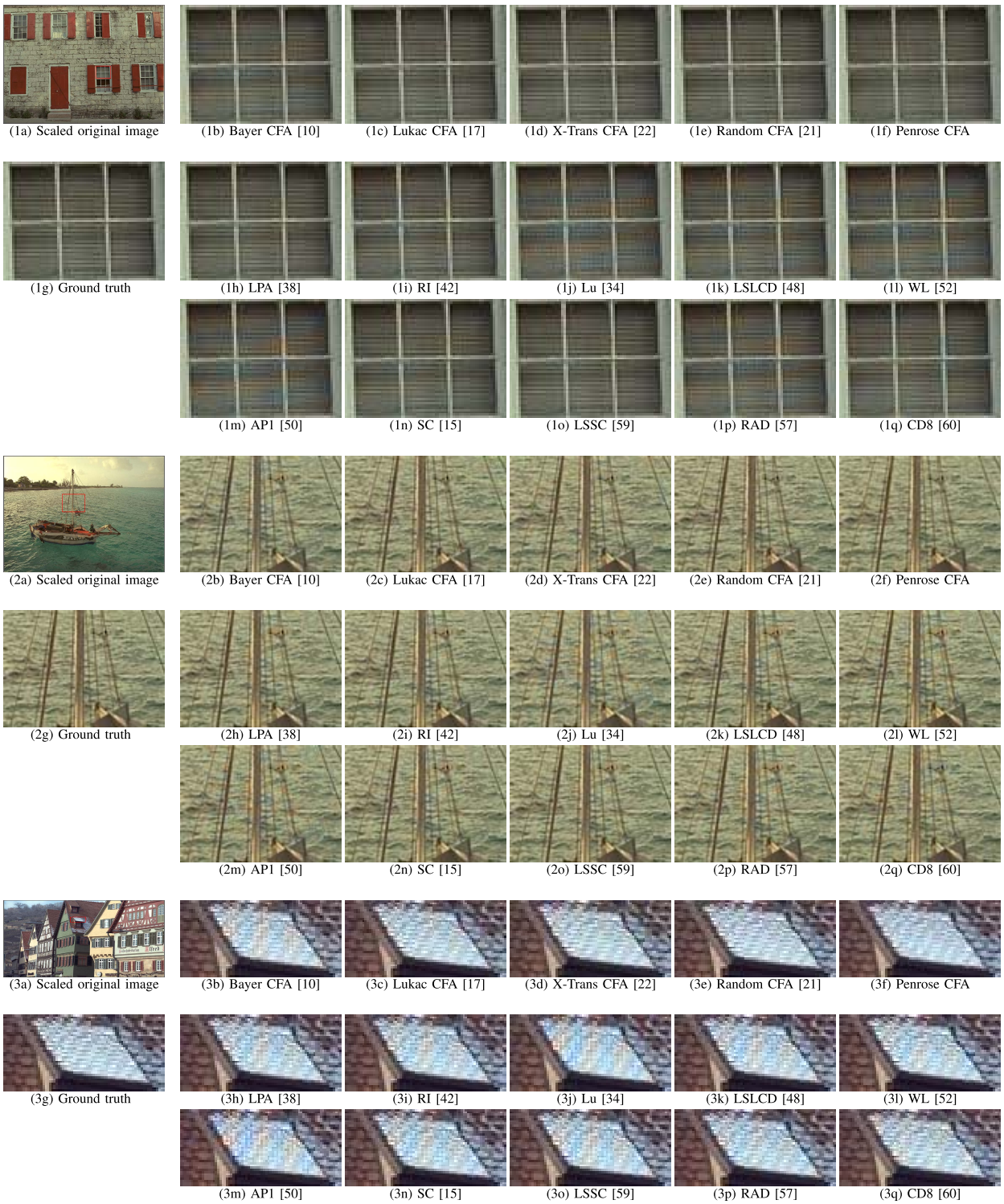


Fig. 6. Blowups of some demosaicked images in the Kodak dataset. From top to bottom, the images are from #1, #6, and #8 Images of the Kodak dataset, respectively. In each group, (a) is the scaled original image, in which the red rectangle indicates the selected patch to blow up; (b)-(e) and (f) are the images demosaicked from raw images by regular RGB CFAs and Penrose CFA, respectively; (g) is the ground truth; (h)-(q) are the images demosaicked by the state-of-the-art demosaicking methods on the Bayer CFA [10]. From all the three groups of images, we can clearly see that the images demosaicked from raw images by Bayer CFA [10], Lukac CFA [17], and X-Trans CFA [22] have severe false color artifacts at the window (false blue and yellow color artifacts), around the mast (false blue and yellow color artifacts), and on the roof of skylight (color stripe in the snow-covered area), while those by Random CFA [21] and Penrose CFA have better subjective quality.

especially in removing false color at highly textured regions or along edges. Although the examples in Fig. 6 seem to show that Random CFA is comparable to Penrose CFA, Fig. 7 and 8 show that Random CFA can still result in false color along highly contrasted edges (Please read the captions for the descriptions on visual difference.).

Fig. 7. More blowups of demosaicked images in the Kodak dataset. From top to bottom, the images are from #1, #6, 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)-(e) and (f) are the images demosaicked from raw images by regular RGB CFAs and Penrose CFA, respectively; (g) is the ground truth; (h)-(q) are the images demosaicked by the state-of-the-art demosaicking methods on the Bayer CFA [10]. We can observe that the Penrose demosaicked images are almost perfect, while all other images have false color.

The above comparisons validate the advantage of Penrose CFA and also testify that the irregularity of Penrose CFA is important for producing demosaicked images with high visual quality.

C. Comparison With Demosaicking on the Bayer CFA

Both the CFA and the demosaicking method affect the visual quality of the reconstructed full color images. Accordingly, to further demonstrate the benefit of Penrose CFA in improving

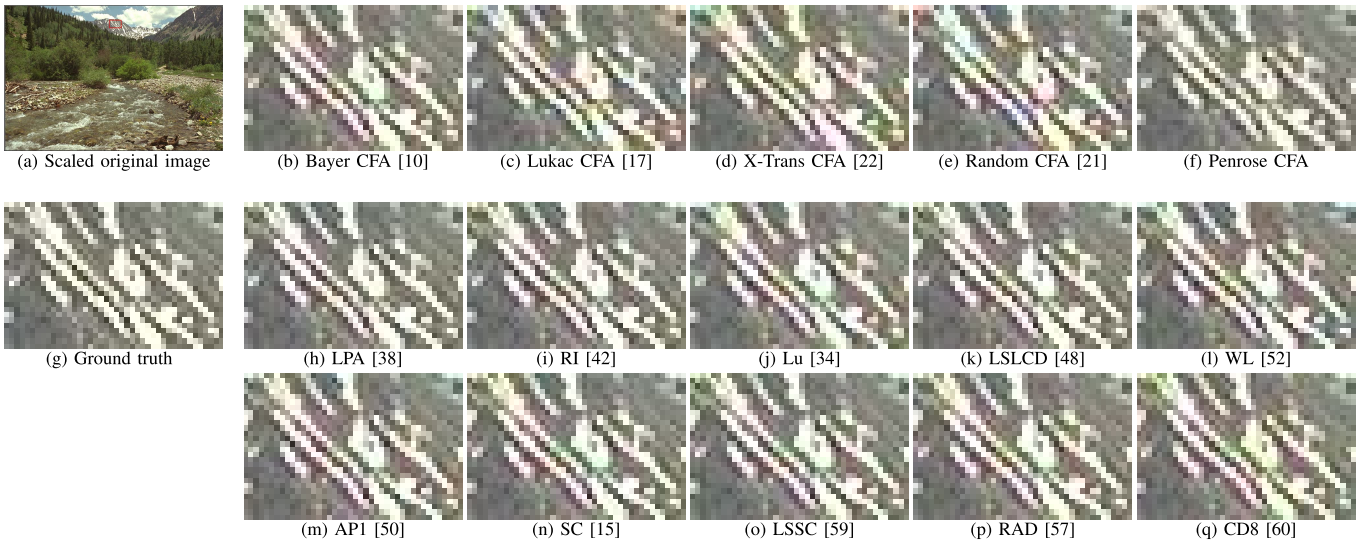


Fig. 8. One more blowup of demosaicked image in the Kodak dataset. (a) is the scaled original image from #13 Image in the Kodak dataset, in which the red rectangle indicates the selected patch to blow up; (b)-(e) and (f) are the images demosaicked from raw images by regular RGB CFAs and Penrose CFA, respectively; (g) is the ground truth; (h)-(q) are the images demosaicked by the state-of-the-art demosaicking methods on the Bayer CFA [10]. We can observe that the Penrose demosaicked image is almost perfect, while all other images have false color.

visual quality, we also present the results of the state-of-the-art demosaicking methods for the Bayer CFA on the Kodak dataset. Among these algorithms, LPA [38] and RI [42] are directional interpolation methods, Lu and Tan [34] is a heuristic method, LSLCD [48] is a frequency domain method, WL [52] and AP1 [50] are wavelet-based methods, SC [15], LSSC [59], RAD [57], and CD [60] are reconstruction-based methods. For these methods, we directly use the results of each method provided online if they exist. Otherwise, we use the codes that are publicly available [79], with default parameters set by their respective authors.

From the right part of Table I, we can also see that Penrose demosaicking does not excel in CPSNR but outperforms in S-CIELAB, being only slightly inferior to LSSC [59]. Besides the adverse reason discussed earlier, we have to highlight that LSSC [59] includes an online dictionary learning strategy during the demosaicking process, while Penrose demosaicking uses a fixed dictionary. From the visual comparison shown in Fig. 6-8, we can see that Penrose demosaicking has better visual quality than other demosaicking methods.

V. Conclusions and Future Work

In this paper, we present the Penrose CFA for color image capturing. Since the Penrose CFA is irregular and aperiodic, most of the existing algorithms are not applicable for demosaicking it. We develop a new demosaicking method based on sparse representation. Extensive experiments on standard test images show that the Penrose CFA is advantageous in reconstructing full color images with high visual quality, in particular, removing false color. Our work complements the Penrose pixels for grayscale image super-resolution [5].

Currently, we simply choose the most commonly used RGB color components for the Penrose CFA. We will try other color components, such as using CMY or optimizing the color components as Hao et al. [18] did. Although with our first demosaicking method the Penrose CFA did not excel in CPSNR, this aspect could be continuously improved, like the effort on the Bayer CFA over the past three decades. Possible strategies include adopting an adaptive dictionary as LSSC [59] did, designing a smart graylevel quantization strategy, and incorporating characteristics of the Penrose CFA. We will also consider designing efficient demosaicking methods for the Penrose CFA. Finally, we thank the reviewers for mentioning the real hexagonal CFA with hexagonal pixels (the one shown in Fig. 2(c) is not a real hexagonal CFA despite its name). To our best knowledge, it is still non-existent. We will compare the Penrose CFA and the real hexagonal CFA thoroughly.

Acknowledgment

The authors would like to thank the Associate Editor and reviewers for their constructive comments and suggestions. They would also like to thank Hongyang Zhang for checking our algorithm and Bo Liu for helping the experiments.

References

[1] L. Middleton and J. Sivaswamy, *Hexagonal Image Processing: A Practical Approach*. New York, NY, USA: Springer-Verlag, 2006.
[2] L. Middleton and J. Sivaswamy, "Edge detection in a hexagonal-image processing framework," *Image Vis. Comput.*, vol. 19, no. 14, pp. 1071–1081, 2001.
[3] R. H. H. Kröger, "Anti-aliasing in image recording and display hardware: Lessons from nature," *J. Opt. A, Pure Appl. Opt.*, vol. 6, no. 8, pp. 743–748, 2004.
[4] B. Grunbaum and G. C. Shephard, *Tilings and Patterns*. New York, NY, USA: Freeman, 1987.
[5] M. Ben-Ezra, Z. Lin, B. Wilburn, and W. Zhang, "Penrose pixels for super-resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 7, pp. 1370–1383, Jul. 2011.
[6] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosaicking: Color filter array interpolation," *IEEE Signal Process. Mag.*, vol. 22, no. 1, pp. 44–54, Jan. 2005.

[7] X. Li, B. Gunturk, and L. Zhang, "Image demosaicing: A systematic survey," *Proc. SPIE, Vis. Commun. Image Process.*, vol. 6822, p. 68221J, Jan. 2008.

[8] D. Menon and G. Calvagno, "Color image demosaicking: An overview," *Signal Process., Image Commun.*, vol. 26, nos. 8–9, pp. 518–533, 2011.

[9] T. Sibley and S. Wagon, "Rhombic Penrose tilings can be 3-colored," *Amer. Math. Monthly*, vol. 107, no. 3, pp. 251–253, 2000.

[10] B. E. Bayer, "Color imaging array," U.S. Patent 3 971 065, Jul. 20, 1976.

[11] J. Couillaud, A. Horé, and D. Ziou, "Nature-inspired color-filter array for enhancing the quality of images," *J. Opt. Soc. Amer. A*, vol. 29, no. 8, pp. 1580–1587, 2012.

[12] J. Couillaud, A. Horé, and D. Ziou, "Improving image acquisition: A fish-inspired solution," in *Advanced Concepts for Intelligent Vision Systems*. Berlin, Germany: Springer-Verlag, 2012, pp. 132–141.

[13] R. D. Fernald, "Chromatic organization of a cichlid fish retina," *Vis. Res.*, vol. 21, no. 12, pp. 1749–1753, 1981.

[14] F. Aurenhammer, "Voronoi diagrams—A survey of a fundamental geometric data structure," *ACM Comput. Surv.*, vol. 23, no. 3, pp. 345–405, 1991.

[15] 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.

[16] C. A. Curcio, K. R. Sloan, Jr., O. Packer, A. E. Hendrickson, and R. E. Kalina, "Distribution of cones in human and monkey retina: Individual variability and radial asymmetry," *Science*, vol. 236, no. 4801, pp. 579–582, 1987.

[17] 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.

[18] 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.

[19] W. Zhu, K. Parker, and M. A. Kriss, "Color filter arrays based on mutually exclusive blue noise patterns," *J. Vis. Commun. Image Represent.*, vol. 10, no. 3, pp. 245–267, 1999.

[20] M. Parmar and S. J. Reeves, "A perceptually based design methodology for color filter arrays," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 3. May 2004, pp. III-473–III-476.

[21] L. Condat, "Color filter array design using random patterns with blue noise chromatic spectra," *Image Vis. Comput.*, vol. 28, no. 8, pp. 1196–1202, 2010.

[22] *X-Trans CMOS*. [Online]. Available: http://www.fujifilmusa.com/products/digital_cameras/x/fujifilm_x_pro1/features/, accessed Nov. 10, 2014.

[23] J. Adams, K. Parulski, and K. Spaulding, "Color processing in digital cameras," *IEEE Micro*, vol. 18, no. 6, pp. 20–30, Nov./Dec. 1998.

[24] *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_Archive/200307/03-029E/, accessed Dec. 1, 2013.

[25] J. F. Hamilton, Jr., and J. T. Compton, "Processing color and panchromatic pixels," U.S. Patent 8 274 715, Sep. 25, 2012.

[26] 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.

[27] A. Chakrabarti, W. T. Freeman, and T. Zickler, "Rethinking color cameras," in *Proc. IEEE Int. Conf. Comput. Photogr. (ICCP)*, May 2014, pp. 1–8.

[28] D. R. Cok, "Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal," U.S. Patent 4 642 678, Feb. 10, 1987.

[29] R. Kimmel, "Demosaicing: Image reconstruction from color CCD samples," *IEEE Trans. Image Process.*, vol. 8, no. 9, pp. 1221–1228, Sep. 1999.

[30] J. E. Adams, Jr., "Interactions between color plane interpolation and other image processing functions in electronic photography," in *Proc. IS&T/SPIE's Symp. Electron. Imag., Sci. Technol.*, 1995, pp. 144–151.

[31] J. E. Adams, Jr., "Design of practical color filter array interpolation algorithms for digital cameras," in *Proc. Electron. Imag.*, 1997, pp. 117–125.

[32] R. Kakarala and Z. Baharav, "Adaptive demosaicing with the principal vector method," *IEEE Trans. Consum. Electron.*, vol. 48, no. 4, pp. 932–937, Nov. 2002.

[33] R. H. Hibbard, "For processing a digitized image signal," U.S. Patent 5 382 976, Jan. 17, 1995.

[34] W. Lu and Y.-P. Tan, "Color filter array demosaicking: New method and performance measures," *IEEE Trans. Image Process.*, vol. 12, no. 10, pp. 1194–1210, Oct. 2003.

[35] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm," *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 360–369, Mar. 2005.

[36] X. Zhang and B. A. Wandell, "A spatial extension of CIELAB for digital color-image reproduction," *J. Soc. Inf. Display*, vol. 5, no. 1, pp. 61–63, 1997.

[37] L. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-error estimation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2167–2178, Dec. 2005.

[38] 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.

[39] D. Menon, S. Andriani, and G. Calvagno, "Demosaicing with directional filtering and *a posteriori* decision," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 132–141, Jan. 2007.

[40] J. S. J. Li and S. Randhawa, "Color filter array demosaicking using high-order interpolation techniques with a weighted median filter for sharp color edge preservation," *IEEE Trans. Image Process.*, vol. 18, no. 9, pp. 1946–1957, Sep. 2009.

[41] 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, 2011.

[42] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Residual interpolation for color image demosaicking," in *Proc. 20th IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 2304–2308.

[43] I. Pekkucuksen and Y. Altunbasak, "Gradient based threshold free color filter array interpolation," in *Proc. 17th IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2010, pp. 137–140.

[44] D. Alleysson, S. Susstrunk, and J. Hérault, "Linear demosaicing inspired by the human visual system," *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 439–449, Apr. 2005.

[45] E. Dubois, "Frequency-domain methods for demosaicking of Bayer-sampled color images," *IEEE Signal Process. Lett.*, vol. 12, no. 12, pp. 847–850, Dec. 2005.

[46] N.-X. Lian, L. Chang, Y.-P. Tan, and V. Zagorodnov, "Adaptive filtering for color filter array demosaicking," *IEEE Trans. Image Process.*, vol. 16, no. 10, pp. 2515–2525, Oct. 2007.

[47] K.-L. Chung, W.-J. Yang, W.-M. Yan, and C.-C. Wang, "Demosaicing of color filter array captured images using gradient edge detection masks and adaptive heterogeneity-projection," *IEEE Trans. Image Process.*, vol. 17, no. 12, pp. 2356–2367, Dec. 2008.

[48] B. Leung, G. Jeon, and E. Dubois, "Least-squares luma–chroma demultiplexing algorithm for Bayer demosaicking," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 1885–1894, Jul. 2011.

[49] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," *IEEE Trans. Image Process.*, vol. 11, no. 9, pp. 997–1013, Sep. 2002.

[50] Y. M. Lu, M. Karzand, and M. Vetterli, "Demosaicking by alternating projections: Theory and fast one-step implementation," *IEEE Trans. Image Process.*, vol. 19, no. 8, pp. 2085–2098, Aug. 2010.

[51] X. Li, "Demosaicing by successive approximation," *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 370–379, Mar. 2005.

[52] D. Menon and G. Calvagno, "Demosaicing based onwavelet analysis of the luminance component," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2. Sep./Oct. 2007, pp. II-181–II-184.

[53] D. H. Brainard, "Bayesian method for reconstructing color images from trichromatic samples," in *Proc. 47th Annu. Conf. Soc. Imag. Sci. Technol.*, vol. 2. 1994, pp. 375–380.

[54] J. Mukherjee, R. Parthasarathi, and S. Goyal, "Markov random field processing for color demosaicing," *Pattern Recognit. Lett.*, vol. 22, nos. 3–4, pp. 339–351, 2001.

[55] D. Keren and M. Osadchy, "Restoring subsampled color images," *Mach. Vis. Appl.*, vol. 11, no. 4, pp. 197–202, 1999.

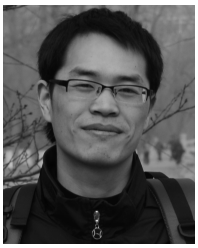
[56] T. Saito and T. Komatsu, "Demosaicing approach based on extended color total-variation regularization," in *Proc. 15th IEEE Int. Conf. Image Process. (ICIP)*, 2008, pp. 885–888.

[57] D. Menon and G. Calvagno, "Regularization approaches to demosaicking," *IEEE Trans. Image Process.*, vol. 18, no. 10, pp. 2209–2220, Oct. 2009.

[58] A. Buades, B. Coll, J.-M. Morel, and C. Sbert, "Self-similarity driven color demosaicking," *IEEE Trans. Image Process.*, vol. 18, no. 6, pp. 1192–1202, Jun. 2009.

[59] 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.*, Sep./Oct. 2009, pp. 2272–2279.
[60] 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.
[61] L. Condat, "A generic proximal algorithm for convex optimization—Application to total variation minimization," *IEEE Signal Process. Lett.*, vol. 21, no. 8, pp. 985–989, Aug. 2014.
[62] Z. Lin and H.-Y. Shum, "Fundamental limits of reconstruction-based superresolution algorithms under local translation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 83–97, Jan. 2004.
[63] R. Lukac and K. N. Plataniotis, "Universal demosaicking for imaging pipelines with an RGB color filter array," *Pattern Recognit.*, vol. 38, no. 11, pp. 2208–2212, 2005.
[64] A. Hore and D. Ziou, "An edge-sensing generic demosaicing algorithm with application to image resampling," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3136–3150, Nov. 2011.
[65] 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.
[66] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.
[67] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Advances in Neural Information Processing Systems*, Red Hook, NY, USA: Curran Associates, 2011, vol. 24, pp. 612–620.
[68] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.
[69] T.-T. Lu and S.-H. Shiou, "Inverses of 2×2 block matrices," *Comput. Math. Appl.*, vol. 43, no. 1, pp. 119–129, 2002.
[70] M. McClure, "Digraph self-similar sets and aperiodic tilings," *Math. Intell.*, vol. 24, no. 2, pp. 33–42, 2002.
[71] S. Wagon, *Mathematica in Action: Problem Solving Through Visualization and Computation*. New York, NY, USA: Springer-Verlag, 2010.
[72] M. McClure, "A stochastic cellular automaton for three-coloring Penrose tiles," *Comput. Graph.*, vol. 26, no. 3, pp. 519–524, 2002.
[73] N. G. de Bruijn, "Algebraic theory of Penrose's non-periodic tilings of the plane. I," *Indagationes Math. (Proc.)*, vol. 84, no. 1, pp. 39–52, 1981.
[74] *Kodak Lossless True Color Image Suite*. [Online]. Available: http://r0k.us/graphics/kodak/, accessed Dec. 1, 2013.
[75] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 689–696.
[76] 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.
[77] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at signal fidelity measures," *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 98–117, Jan. 2009.
[78] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[79] *Reproduce Research in Computational Science*. [Online]. Available: http://www.csee.wvu.edu/~xinl/source.html, accessed Dec. 1, 2013.

Chenyan Bai received the B.E. and M.E. degrees in computer science from the School of Mathematics and Computer, Hebei University, China, in 2008 and 2011, respectively. She is currently pursuing the Ph.D. degree with the School of Computer and Information Technology, Beijing Jiaotong University. Her research interest is image processing.



Jia Li received the bachelor's degree in mathematics and the master'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.

Zhouchen Lin (M'00–SM'08) received the Ph.D. degree in applied mathematics from Peking University, in 2000. He was a Guest Professor with Shanghai Jiao Tong University, Beijing Jiaotong University, and Southeast University. He was also a Guest Researcher with the Institute of Computing Technology, Chinese Academic of Sciences. He is currently a Professor with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University. He is also the Chair Professor with Northeast Normal 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 the *International Journal of Computer Vision*.

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

Yen-Wei Chen received the B.E. degree from Kobe University, Kobe, Japan, in 1985, and the M.E. and D.E. degrees from Osaka University, Osaka, Japan, in 1987 and 1990, respectively. He is currently a Full Professor with the College of Information Science and Engineering, and the Director of the Research Center of Advanced ICT for Medical and Healthcare with Ritsumeikan University, Japan. He is the Chair Professor with Zhejiang University, China. His research interests include medical image analysis, pattern recognition, and image processing. He has received many distinguished awards, including the 2012 International Conference on Pattern Recognition (ICPR) Best Scientific Paper Award, and the Outstanding Chinese Oversea Scholar Fund of the Chinese Academy of Science. He has authored over 300 research papers in a number of leading journals and leading conferences, including the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, *Pattern Recognition*, the Conference on Medical Image Computing and Computer Assisted Intervention, ICPR, and the International Conference on Image Processing.