

Universal Demosaicking of Color Filter Arrays

Chao Zhang, Yan Li, Jue Wang, and Pengwei Hao, *Member, IEEE*

Abstract—A large number of color filter arrays (CFAs), periodic or aperiodic, have been proposed. To reconstruct images from all different CFAs and compare their imaging quality, a universal demosaicking method is needed. This paper proposes a new universal demosaicking method based on inter-pixel chrominance capture and optimal demosaicking transformation. It skips the commonly used step to estimate the luminance component at each pixel, and thus avoids the associated estimation error. Instead, we directly use the acquired CFA color intensity at each pixel as an input component. Two independent chrominance components are estimated at each pixel based on the inter-pixel chrominance in the window, which is captured with the difference of CFA color values between the pixel of interest and its neighbors. Two mechanisms are employed for the accurate estimation: distance-related and edge-sensing weighting to reflect the confidence levels of the inter-pixel chrominance components, and pseudoinverse-based estimation from the components in a window. Then from the acquired CFA color component and two estimated chrominance components, the three primary colors are reconstructed by a linear color transform, which is optimized for the least transform error. Our experiments show that the proposed method is much better than other published universal demosaicking methods.

Index Terms—Demosaicking, color filter array, linear combination, matrix pseudo-inverse

I. INTRODUCTION

ALMOST every modern camera, video camera, machine vision system and scanner has a mosaicked color filter array (CFA) fabricated on top of the light sensors to capture color images. CFA measures the energy of only one color light at each pixel, and a technique known as “demosaicking” is then used to reconstruct all three primary colors for each pixel, such as red, green and blue [10,21,33,34]. For high-quality image formation, we need both an optimal CFA and an optimal associated demosaicking method.

Manuscript received September 17, 2013. This work was supported in part by the National Key Basic Research Project of China (973 Program) under Grant 2015CB352303 and Grant 2011CB302400 and in part by the National Natural Science Foundation (NSF) of China under Grant 61071156 and Grant 61131003.

C. Zhang is with Key Laboratory of Machine Perception (MOE), School of Electronic Engineering and Computer Science, Peking University, Beijing, 100871, China. (phone: 0086-10-6275-7000; fax: 0086-10-6275-9444; e-mail: chzhang@cis.pku.edu.cn).

Y. Li is with Lely Technologies, Weverskade 110, 3147 PA, Maassluis, the Netherlands. (e-mail: yanli.grass@gmail.com)

J. Wang is with the School of Electronic Engineering and Computer Science, Peking University, 100871, China (e-mail: cn.wangjue@pku.edu.cn).

P. Hao is with the School of Electronic Engineering and Computer Science, Queen Mary, University of London, E1 4NS, UK (phone: 0044-20-7882-5207, fax: 0044-20-7882-7997, e-mail: p.hao@qmul.ac.uk).

Except for some random CFAs, a two-dimensional CFA usually comprises a plurality of a minimal repeating pattern tiled to cover the entire array of image sensors. Different minimal repeating CFA patterns characterize different periodic CFAs. The first CFA design was inspired by the human retinal mosaic and is named after its inventor, Bryce E. Bayer of Eastman Kodak [3]. Bayer CFA uses the minimal repeating pattern of size 2×2 with 1 red, 1 blue and 2 green pixels, and it is the most popular CFA on consumer digital cameras. Alternatives include the CYGM filter (cyan, yellow, green, magenta) of Kodak [12], the RGBE filter (red, green, blue, emerald) of Sony [32] and the 45-degree rotated Bayer filter known as SuperCCD of Fujifilm [29], whose minimal repeating patterns are all of size 2×2 . The recent trends in commercial CFA designs are to increase the sensitivity to light by using panchromatic pixels (or white pixels) [15,28,31] and to provide better resistance to color moiré patterns by using less-regular or random pixel arrangements [30,37].

Periodic CFAs were found to have elegant representations in the frequency domain by using the symbolic discrete Fourier transform (DFT), in which there is one luminance component at zero frequency multiplexed with a few chrominance components at high frequencies [2,7,11,16,17,35,36,38]. Based on such representations, the analysis, design and demosaicking of CFAs become easier and more accurate. In comparison, a random CFA helps minimize the moiré artifacts and remove the anti-aliasing filter in camera, using which actually reduces spatial resolution [30]. However, a random CFA cannot be periodically generated, its corresponding DFT version does not have so obvious and simple chrominance components, and we cannot directly and efficiently use the demosaicking methods specialized for periodic CFAs.

The demosaicking methods for periodic CFAs are relatively well investigated and better optimized, e.g. for Bayer CFA [7,38], but for random CFAs a good demosaicking method has not been fully and systematically explored yet. Generally a specific demosaicking method is specialized for a particular CFA pattern, which is not applicable to other CFAs. It would be very beneficial to have a universal demosaicking method applicable to all CFAs, periodic or aperiodic, regular or irregular, uniform or nonuniform. A well-performed universal method also provides an important platform for comparison between different CFAs and to give feedbacks in designing new CFAs. The ultimate aim of this paper is to design an optimal method for universal demosaicking of all CFAs.

Before summarizing the previous universal demosaicking methods, we first introduce the three aspects of information that can be exploited to make demosaicking better: spatial correlation, spectral correlation and edge information (or

spatial discontinuity). a) Except for a small number of edge pixels, most areas of natural images are smooth, or spatially highly correlated. As the color values are very similar within a small homogeneous region in an image, such spatial correlation helps greatly in demosaicking [24]. Spatial correlation can be explored by spatial filtering which uses more information from closer pixels or spatial transformations such as the discrete Fourier transform and the discrete wavelet transform. b) Spectral dependency/correlation has also been frequently exploited for better demosaicking [10,21,26,43]. The highly-correlated spectra of illuminance and reflectance in the natural world lead to high correlation between different color planes of an image. As a result, chrominance is highly band-limited and chrominance interpolation is more accurate than direct individual color interpolation for demosaicking. c) Edges or contours are significant in human vision since, due to the visual characteristics of the Mach band effect, the human vision system is very sensitive to the high spatial frequency information in images. Consequently, the demosaicking error around edges is easily noticeable and annoying. As edges may appear anywhere in any direction in an image and have various forms, sharp or soft, strong or weak, long or short, flush or jagged, continuous or fragmental, an edge-sensitive demosaicking method needs to be adaptive and non-linear. In summary, a good demosaicking method should exploit both the spatial and the spectral correlations of a color image and should also consider the discontinuity in the image, such as the edge information [1,5,9, 13,18,20,22,27,39-45].

A big number of demosaicking methods have been created for a few specific CFAs, but few of such methods may be adaptable to other CFAs. For universal demosaicking methods that can be used for all CFAs, so far, only a few have been proposed and most of the methods are still for periodic CFAs. The first universal demosaicking method was presented by Lukac and Plataniotis [19], which first estimates the missing pixel colors for the most populated color (green), then finds the other missing colors with the weighted mean chrominance in a neighborhood, and finally updates the estimated missing colors with the weighted mean chrominance as a post-processing. The method uses the absolute difference of pixel values of the same color as the structural information for the weights, which makes the method nonlinear and image-dependent. Menon and Calvagno solved the demosaicking problem by a Tikhonov regularized optimization [20]. It estimates all the three primary colors at each pixel directly from the acquired CFA-filtered pixel values. The regularization was realized by imposing three smoothness constraints, involving individual color channels, chrominance and horizontal and vertical edges. Condat proposed an approach to minimize a variational functional under the constraint of consistency and implemented an iterative method to maximize the smoothness by minimizing the high-frequency energy of luminance and chrominance components [6]. Gu et al published another iterative method by using filterbank transforms [8]. Generally the iterativeness makes a method take longer time to converge and mostly slower than direct methods, and the convergence speed is also highly dependent on the initial estimate, although this becomes

less a problem as better and faster methods being proposed in the field. Condat later proposed a generic proximal algorithm for faster optimization [50], but it is still iterative and the image quality is not improved substantially. Hore and Ziou in [14] added an edge-sensing mechanism to the universal method in [19] and unanimously improved the demosaicking performance with all the seven uniform CFAs they tested. Another universal demosaicking was proposed by Singh and Singh [48], which is non-iterative and using space-variant filters for random CFAs. The method is based on the generalized inverse of the matrix for CFA sampling relations of primary colors. To make the method perform better, it employs some nonlinear image enhancement methods, but the improvement seems to be limited. Later, they better optimized the linear part of the method but only for regular patterns [23] because it is based on the frequency domain representations [11]. The methods based on transforms generally implicitly take CFAs as being regular and uniform, and the methods with the discrete Fourier transform are generally applied to regular, uniform and periodic CFAs, since the discrete Fourier transform generally does not map irregular, nonuniform or aperiodic signals into impulses at some frequency points, which makes very difficult to represent and analyze CFAs in the frequency domain, not so easy as for the periodic CFAs as shown in [11]. A recent publication is for a small class of CFAs with quincuncial white pixels [46]. It utilizes the most populated and more accurately interpolated white pixel channel to improve the channels of the primary colors that we eventually need. Anyway, it is not a universal demosaicking method that can be applied to all CFAs.

Based on the above brief analysis of the previous work, we know that a universal demosaicking method could work better if it is in the spatial domain and interpolates the chrominance from neighbor pixels. We developed a novel non-iterative universal method based on accurate chrominance estimation in the spatial domain and the optimal demosaicking color transformation. We do not estimate the luminance or any individual color planes for color reconstruction; and the pixel values measured by a CFA are used instead to avoid the estimation error of the luminance component. Two chrominance components at each pixel are estimated with the inter-pixel chrominance in a window, which is captured by the color difference between the adjacent neighbors. Inter-pixel chrominance components are weighted with respect to distance and edge, and then optimally combined for the target chrominance directions. The linear demosaicking color transformation is also optimized for the least error to reconstruct the three primary colors from the chrominance and CFA color at each pixel. The comparison between our method and other two universal demosaicking methods is highly favorable with a few CFA patterns, the Bayer CFA [3], the Kodak's CFA2.0 [15], Fuji X-Trans [30], Sony RGBW [31], a random CFA [37] and a high-light sensitivity RGBW CFA proposed previously by us [25]. For brevity of notation, we refer to our adaptive chrominance-based universal demosaicking method as ACUDe.

The rest of this paper is organized as follows. Section II first gives an overview of the proposed method, whose six main

steps are described in detail in Sections III to VI, The demosaicking results are presented in Section VII. Finally, Section VIII concludes the paper.

II. OVERVIEW OF THE PROPOSED METHOD

A. The idea

In the CIE colorimetry system, a color visible to human vision can be represented with three primary colors, such as red, green and blue (RGB) in the additive color space for light color mixture, or cyan, magenta and yellow (CMY) in the subtractive space for pigment color mixture. Accordingly, we can decompose an incident image and CFA coefficients with three primary colors. Suppose that the CFA color coefficients at pixel (x, y) are $\mathbf{f}(x, y) = [r(x, y), g(x, y), b(x, y)]^T = [r, g, b]^T(x, y)$, or simply $\mathbf{f} = [r, g, b]^T$ if there is no ambiguity. Similarly the incident image can be represented as $\mathbf{i}(x, y) = [R(x, y), G(x, y), B(x, y)]^T = [R, G, B]^T(x, y)$ or $\mathbf{i} = [R, G, B]^T$. The acquired pixel values with the CFA can be expressed as the dot product of their corresponding three-dimensional vectors: $C_f(x, y) = r(x, y)R(x, y) + g(x, y)G(x, y) + b(x, y)B(x, y) = \mathbf{f}^T(x, y) \cdot \mathbf{i}(x, y)$, or $C_f = \mathbf{f}^T \cdot \mathbf{i}$. Assume that all the coefficients, r, g, b, R, G, B , are in the range of $[0, 1]$, then the CFA response should reach its highest $C_f = 1$ to the highest incident white light (fully saturated white $[R, G, B] = [1, 1, 1]$), so we have $r + g + b = 1$.

The difference of two CFA responses $C_{f1} - C_{f2}$ is the so-called color chrominance. Similarly it can be expressed with three primary colors as $C_{h1} = \mathbf{h}_1^T \cdot \mathbf{i} = p_r R + p_g G + p_b B$, where $\mathbf{h}_1 = [p_r, p_g, p_b]^T$. The chrominance should be zero if $R = G = B$, so this always holds: $p_r + p_g + p_b = 0$. Vector \mathbf{h}_1 can be seen as the hue or direction of the chrominance, and C_{h1} as its saturation or amplitude in the direction. With a different set of coefficients for the vector, $\mathbf{h}_2 = [q_r, q_g, q_b]^T$ we can have the chrominance in another direction: $C_{h2} = \mathbf{h}_2^T \cdot \mathbf{i} = q_r R + q_g G + q_b B$, where $q_r + q_g + q_b = 0$. The vectors \mathbf{h}_1 and \mathbf{h}_2 are in 3D and always satisfies the equations $\mathbf{h}_1^T \cdot [1, 1, 1]^T = 0$ and $\mathbf{h}_2^T \cdot [1, 1, 1]^T = 0$, so the intrinsic dimension of chrominance is two.

With one CFA response and two chrominance at a pixel, we can formulate their linear relations in a matrix form:

$$\begin{bmatrix} C_f \\ C_{h1} \\ C_{h2} \end{bmatrix} = \begin{bmatrix} \mathbf{f}^T \\ \mathbf{h}_1^T \\ \mathbf{h}_2^T \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} r & g & b \\ p_r & p_g & p_b \\ q_r & q_g & q_b \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

or by using a multiplexing matrix \mathbf{T} :

$$\begin{bmatrix} C_f \\ C_{h1} \\ C_{h2} \end{bmatrix} = \mathbf{T} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2)$$

Since a fully saturated white point $[R, G, B] = [1, 1, 1]$ should be mapped into a fully saturated CFA response $C_f = 1$ and zero chrominance components, $C_{h1} = 0$ and $C_{h2} = 0$, the following equation holds:

$$[1, 0, 0]^T = \mathbf{T} \cdot [1, 1, 1]^T \quad (3)$$

The matrix \mathbf{T} is invertible if the two chrominance directions are independent of each other, or the rank of the two chrominance rows in \mathbf{T} is 2. Then we can reconstruct the three primary color components by matrix inverse:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \mathbf{T}^{-1} \cdot \begin{bmatrix} C_f \\ C_{h1} \\ C_{h2} \end{bmatrix} \quad (4)$$

We can use a demosaicking matrix \mathbf{D} to represent this inverse transform matrix \mathbf{T}^{-1} , which can be found mathematically as in [11]:

$$\mathbf{D} = \mathbf{T}^{-1} = \begin{bmatrix} 1 & \frac{bq_g - gq_b}{q_g p_r - q_r p_g} & \frac{gp_b - bp_g}{q_g p_r - q_r p_g} \\ 1 & \frac{rq_b - bq_r}{q_g p_r - q_r p_g} & \frac{bp_r - rp_b}{q_g p_r - q_r p_g} \\ 1 & \frac{q_g p_r - q_r p_g}{q_g p_r - q_r p_g} & \frac{q_g p_r - q_r p_g}{q_g p_r - q_r p_g} \end{bmatrix} \quad (5)$$

To reconstruct the three primary colors, three components C_f, C_{h1} and C_{h2} are needed. For C_f , previous methods typically first estimate a general luminance component for the whole image [2,7,11,35,36,38], that is, a component of $C_f^l(x, y) = l(x, y) = r_0 R(x, y) + g_0 G(x, y) + b_0 B(x, y)$ with the same coefficients $[r_0, g_0, b_0]$ for all pixels. One example is to use $[0, 1, 0]$ as the coefficients to estimate the green channel of the whole image first. In comparison, our method does not need this intermediate estimation step and uses directly the measured CFA response instead. By skipping the step of estimating luminance/green channel, our method introduces less error in demosaicking and can lead to better results.

In summary, our idea is briefly outlined as follows: With a specific CFA color $\mathbf{f}(x, y)$, we can find the optimal demosaicking matrix \mathbf{D} , and then the two target chrominance directions, $\mathbf{h}_1(x, y)$ and $\mathbf{h}_2(x, y)$, are given in the multiplexing matrix \mathbf{T} , which is the inverse of \mathbf{D} . With $C_f(x, y)$ as the measured CFA response, we only need estimate two chrominance components, $C_{h1}(x, y)$ and $C_{h2}(x, y)$, i.e. two chrominance amplitudes in the two target chrominance directions for each pixel in our proposed method. Finally, we use the demosaicking matrix to transform the three components back to the three primary colors.

B. The method

The method comprises three phases. The first phase is to find the optimal demosaicking transform matrix and the corresponding chrominance directions for a given CFA color. The second phase concerns generating two sets of weights used to combine inter-pixel chrominance in a window centered at a pixel for the estimation towards the target chrominance directions. The third phase uses the weights generated in the second phase to estimate the two chrominance amplitudes and then reconstructs the primary colors with Eq. (4). The three phases are further divided into six steps, which we describe briefly below, and their details are presented in the following Sections.

Phase 1: To find the optimal transformation (1 step):

Step 1) For a given CFA color, we find the optimal demosaicking transform matrix \mathbf{D}^* by minimizing the 2-norm

of the matrix \mathbf{D} in Eq. (5). Then we can further find the multiplexing matrix \mathbf{T} as in Eq (1) by matrix inverse, and the two lower rows in \mathbf{T} , \mathbf{h}_1^T and \mathbf{h}_2^T , are the corresponding target chrominance directions, which are needed in the next phase (Step 2-4), and the demosaicking matrix is used in the last step (Step 6). This step is only related to the CFA colors, so it can be done offline and in advance. More details are given in Section VI.

Phase 2: Weight generation, divided into three steps (Fig. 1):

Step 2) Inter-pixel chrominance capture: A chrominance at each pixel can be calculated with two color measurements, but we have only one color measurement $C_f(x, y)$ at each pixel in the CFA-filtered image. As measurements of different colors can be found in the neighborhood of the pixel, we use the difference to compute inter-pixel chrominance. Such inter-pixel chrominance is only a rough estimate of the chrominance at a pixel, but we can combine the inter-pixel chrominance in a bigger neighborhood or a window for more accurate estimation of the two chrominance components in the target directions by using the following two steps. If we stack CFA colors in a matrix \mathbf{F} , similar to the matrix representations in [20], this step to capture the differences of colors is equivalent to multiplying a differentiation matrix \mathbf{G} to \mathbf{F} , resulting in $\mathbf{G} \cdot \mathbf{F}$. Details are given in Section III.

Step 3) Weighting: All the captured inter-pixel chrominance components in a window can contribute to the chrominance at its central pixel, but with different weights. One approach to obtain these weights is with a 2D distance-related function: the closer to the central pixel, the higher the weights. With such fixed distance-related weights only, this proposed method is a linear universal demosaicking method. Another approach is to generate edge-sensitive weights, adaptively with low weights for pixels crossing an edge and high weights for pixels in a homogeneous area. These weights are image-dependent and nonlinear, and make the whole method nonlinear and adaptive. Our method combines these two sets of weights as a two-pass algorithm. We first use only the distance-related weighting to obtain an initial estimate, which is then used to find the edge-sensing weights for adaptive demosaicking. Representing these weights as a diagonal matrix, this step is equivalent to a matrix multiplication: $\mathbf{W} \cdot \mathbf{G} \cdot \mathbf{F}$. See Section IV.

Step 4) Linear combination: The directions of the inter-pixel chrominance are usually different from that of the target chrominance \mathbf{h}_1 and \mathbf{h}_2 , and we need to combine them towards the target chrominance directions. One way is to find a vector \mathbf{x} for the linear combination. This vector is usually not unique, and we find it by using the pseudo-inverse of the matrix constructed from the inter-pixel chrominance. The matrix representation of this combination is $\mathbf{h}_k^T = \mathbf{x}_k^T \cdot \mathbf{W} \cdot \mathbf{G} \cdot \mathbf{F}; k = 1, 2$. See Section V.

Phase 3: Demosaicking by weighted sum in two steps (See Fig. 2):

Step 5) Chrominance estimation: For each pixel, two chrominance components C_{h1} and C_{h2} are estimated by weighting over the acquired CFA-filtered image C_f with the chrominance weights generated in Phase 2, $\hat{\mathbf{h}}_1$ and $\hat{\mathbf{h}}_2$. See the

last part of Section V.

Step 6) Demosaicking transformation: This is done by transforming from $[C_f, C_{h1}, C_{h2}]$ to three primary colors $[R, G, B]$ with Eq. (4), by using the demosaicking matrix \mathbf{D}^* obtained in Step 1. It is similar to that in the previously published paper [11], but the transform matrix \mathbf{D} is better optimized for the least demosaicking transform error (in Step 1). See Section VI.

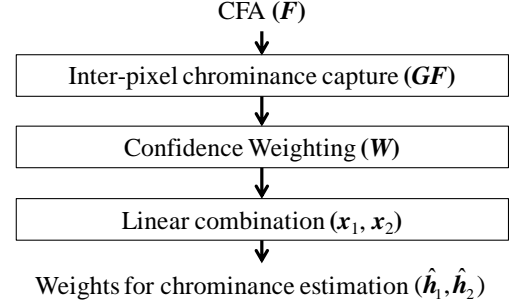


Fig. 1. Phase 2: Weight generation

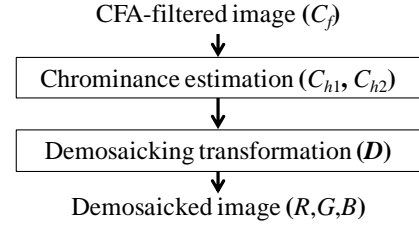


Fig. 2. Phase 3: Demosaicking

III. INTER-PIXEL CHROMINANCE CAPTURE

Two independent chrominance components are needed at each pixel to reconstruct its three colors, but a CFA-filtered image provides only one color $\mathbf{f}(x, y)$ at each pixel (x, y) . So we can only estimate the chrominance at a pixel by using its neighboring pixels $\mathbf{f}(x + dx, y + dy)$, where (dx, dy) are the offsets to the pixel, such as $-1, 0, 1$, and $(dx, dy) \neq (0, 0)$. Therefore, we have to capture the inter-pixel chrominance, which is measured as the difference between a CFA-filtered pixel and its adjacent nearest neighbors:

$$C_g(x, y, dx, dy) = C_f(x + dx, y + dy) - C_f(x, y) = \mathbf{f}^T(x + dx, y + dy) \cdot \mathbf{i}(x + dx, y + dy) - \mathbf{f}^T(x, y) \cdot \mathbf{i}(x, y) \quad (6)$$

If the image is smooth enough at (x, y) , or we have $\mathbf{i}(x + dx, y + dy) \approx \mathbf{i}(x, y)$, then we can have a very rough estimate for the “at-pixel” chrominance $C_g(x, y)$ from the inter-pixel chrominance:

$$C_g(x, y, dx, dy) \approx \{\mathbf{f}(x + dx, y + dy) - \mathbf{f}(x, y)\}^T \cdot \mathbf{i}(x, y) = \mathbf{g}^T(x, y, dx, dy) \cdot \mathbf{i}(x, y) \quad (7)$$

where $\mathbf{g}(x, y, dx, dy) = \mathbf{f}(x + dx, y + dy) - \mathbf{f}(x, y)$ is the difference of CFA colors, which is independent of the image and can be obtained off-line with CFA colors only. Eq. (7) is meaningful only if the two CFA pixel colors are different, $\mathbf{f}(x + dx, y + dy) \neq \mathbf{f}(x, y)$, otherwise it is equal to zero and

does not contribute to the chrominance estimation.

We list the formulae of the CFA color difference and the inter-pixel chrominance amplitude below such that we can see the relationship between them is the same:

$$\begin{cases} C_g(x, y, dx, dy) = C_f(x + dx, y + dy) - C_f(x, y) \\ \mathbf{g}(x, y, dx, dy) = \mathbf{f}(x + dx, y + dy) - \mathbf{f}(x, y) \end{cases} \quad (8)$$

Because of the existence of noise and edges in images, the inter-pixel chrominance calculated with Eq. (7) is usually not very accurate. More accurate at-pixel chrominance can be obtained by linearly combining such inaccurate inter-pixel chrominance at a number of pixels, $C_g(x + s, y + t)$, in a bigger neighborhood or a window, $(s, t) \in \Omega$. The estimation method can be represented as a weighted sum model in the window:

$$C_h(x, y) = \sum_{dx, dy} \sum_{s, t} \hat{h}(x, y, s, t, dx, dy) \cdot C_g(x + s, y + t, dx, dy) \quad (9)$$

If the image around the pixel is smooth and the weights are properly chosen, the estimation can be further approximated as:

$$\begin{aligned} C_h(x, y) &\approx \sum_{dx, dy} \sum_{s, t} \hat{h}(x, y, s, t, dx, dy) \\ &\quad \cdot \mathbf{g}^T(x + s, y + t, dx, dy) \cdot \mathbf{i}(x, y) \\ &= \left\{ \sum_{dx, dy} \sum_{s, t} \hat{h}(x, y, s, t, dx, dy) \right. \\ &\quad \left. \cdot \mathbf{g}^T(x + s, y + t, dx, dy) \right\} \cdot \mathbf{i}(x, y) \end{aligned} \quad (10)$$

Then, we can have an equation for the target chrominance direction, which is related to the CFA colors only:

$$\mathbf{h}(x, y) = \sum_{dx, dy} \sum_{s, t} \hat{h}(x, y, s, t, dx, dy) \cdot \mathbf{g}(x + s, y + t, dx, dy) \quad (11)$$

Reversely, if we can find proper weights for Eq. (11) with CFA colors to make the target chrominance directions, we can then use Eq. (8) and Eq. (9) to estimate the chrominance amplitudes with the acquired CFA-filtered image, which will be given in the next two sections.

To better formulate our method, we use an $N \times 3$ matrix \mathbf{F} to represent the CFA colors $\mathbf{f}(x + s, y + t)$ of all the pixels in the window centered at pixel $\mathbf{f}(x, y)$, where 3 columns are for three primary colors and N is the number of pixels in the window, $(s, t) \in \Omega$. Each CFA pixel with color $\mathbf{f}(x + s, y + t)$ corresponds to one row in \mathbf{F} as $[r, g, b](x + s, y + t)$. Then all the inter-pixel chrominance directions relative to all the (dx, dy) nearest neighbors for all the pixels in the window $\mathbf{g}(x + s, y + t, dx, dy)$ can be represented by a stacked matrix \mathbf{G} multiplied to \mathbf{F} , or $\mathbf{G} \cdot \mathbf{F}$. Each row of \mathbf{G} corresponds to one inter-pixel chrominance direction and has only two non-zero elements: -1 at the position of the center pixel $(x + s, y + t)$ and 1 at the position of its nearest neighbor $(x + s + dx, y + t + dy)$. Multiplying this row to \mathbf{F} gives the inter-pixel chrominance $\mathbf{g}(x + s, y + t, dx, dy)$ as the difference between the CFA color at pixel $(x + s, y + t)$ and that of its (dx, dy) nearest neighbor. So \mathbf{G} is a very sparse matrix with most of the

elements equal to 0 and the nonzero elements equal to -1 or 1 . Such a matrix representation is similar to that in [20].

For a rectangular grid of image pixels, if the window size is $m \times n$, then $N = mn$. If each pixel has M adjacent nearest neighbors, the number of rows of \mathbf{G} and the number of weights \hat{h} can be mnM if a border extension is applied to the window by border pixel duplication, or can be $(m - 2)(n - 2)M$ without a border extension (where a border extension is needed only on the borders of the image).

By multiplying \mathbf{G} to \mathbf{F} , we can capture all the inter-pixel chrominance directions in the window. Comparing Eqs. (8,9,11), we know that, if we use a vector of all CFA-filtered pixel values in the window to replace \mathbf{F} , we can actually capture all the inter-pixel chrominance amplitudes in the corresponding directions in the window.

IV. CONFIDENCE WEIGHTING

The inter-pixel chrominance components are captured with neighbor CFA pixels, and all the components in the window can be used for the linear combination towards the target chrominance directions. However, some inter-pixel chrominance components with higher confidence levels can contribute more to the target than others, or less otherwise. This is achieved by weighting. Considering the confidence levels of the captured inter-pixel chrominance, the weighting involves two aspects of information: (a) distance-related: the closer, the higher; (b) edge-related: the smoother, the higher.

A. Distance-related weights

It has been shown in [2,10] that the chrominance information in color images is mostly of very low spatial frequency, and statistically distributes in a diamond-shaped region symmetrically around frequency 0. So the chrominance information changes infrequently in the spatial domain and can be reasonably estimated from inter-pixel chrominance. For the inter-pixel chrominance, the closer to the pixel of interest, the more similar to the at-pixel chrominance. Therefore, inter-pixel chrominance components closer to the pixel of interest, at which we wish to estimate the target chrominance, should be given higher weights.

One way to assign such distance-related weights is to use a function of only the distance $d(s, t)$ between the weighted point $(x + s, y + t)$ and the central pixel of interest (x, y) , ignoring the directions that the inter-pixel chrominance is captured (dx, dy) :

$$w_d(x, y, s, t, dx, dy) = w_d(s, t) = w(d(s, t)) \quad (12)$$

The distance function $d(s, t)$ can be a p -norm distance including Euclidean distance (2-norm) :

$$d(s, t) = \|(s, t)\|_p = \sqrt[p]{|s|^p + |t|^p} \quad (13)$$

The weighting function $w(d)$ can be a Gaussian function, a raised cosine function, or a function like a low-pass filter, such as Dubois's piecewise-linear filter [7]. The weighting with a Gaussian function is:

$$w(d) = \exp\left(-\frac{d^2}{2d_c^2}\right), \quad (14)$$

where d_c is a positive constant for the cut-off distance, $d = d(s, t)$ is the distance function. Our experiment shows that the performance of different weighting functions is similar but the piecewise linear function is more flexible to generate weights.

Such distance-related weighting is independent of images, so the method is linear if no further adaptive weighting is applied.

B. Edge-sensing weights

If there is an edge near the pixel of interest, the inter-pixel chrominance estimated across the edge is not so reliable as those estimated along the edge or away from edges in a smooth neighborhood. For those inter-pixel chrominance across an edge, the confidence level should be lower than the others. Considering the confidence level of the captured inter-pixel chrominance, we introduce edge-sensing weights to relate weights to edges. Such weighting is image-dependent, thus the corresponding demosaicking method is adaptive and nonlinear.

To give appropriate edge-sensing weights, we need find the edges in the color mosaicked image first. This is mostly done with the luminance image, which is not immediately ready to use. However, we can have an initial rough estimate without using edge-related weights for the color image $\hat{i}(x, y)$ or the luminance $\hat{l}(x, y)$. This rough estimation is actually linear universal demosaicking with distance-related weighting only.

At every pixel (x, y) , to find the edge-sensing weight applied to the captured inter-pixel chrominance, $C_g(x, y, dx, dy)$, we can estimate all the directional gradients from the pixel to all its neighbor pixels:

$$e(x, y, dx, dy) = \hat{l}(x + dx, y + dy) - \hat{l}(x, y) \quad (15)$$

The mean edge energy at each pixel can be estimated as:

$$E^2(x, y) = \frac{1}{M(x, y)} \sum_{dx, dy} e^2(x, y, dx, dy) \quad (16)$$

where $M(x, y)$ is the number of the neighbor pixels at (x, y) .

Then the weights for all the neighbor pixels at each pixel of interest can be roughly estimated as $1 - e_r$, where

$$e_r(x, y, dx, dy) = \frac{e^2(x, y, dx, dy)}{2E^2(x, y)} \quad (17)$$

Considering the estimation error and the sensitivity to noise, we use the following piecewise function for the edge-related weights:

$$w_e(x, y, dx, dy) = \begin{cases} 1 & \sigma < \varepsilon_1 \text{ or } e_r < \varepsilon_2 \\ 0 & e_r > \varepsilon_3 \\ \frac{\varepsilon_3 - e_r}{\varepsilon_3 - \varepsilon_2} & \text{otherwise} \end{cases} \quad (18)$$

where $\sigma = E^2(x, y) / \max_{x, y} E^2(x, y)$, $0 \leq \varepsilon_1 \leq 5\%$, $0 \leq \varepsilon_2 \leq 1/2$ and $1/2 \leq \varepsilon_3 \leq 3/2$ are three cut-off numbers. To our experience, these numbers can be typically: $\varepsilon_1 = 0.5\%$, $\varepsilon_2 = 0$, and $\varepsilon_3 = 1$ if $M \geq 4$.

The above weighting is similar to bilateral filtering [49], but with a different similarity function which is measured only with the luminance values of the neighbors. The distance-related weights can be taken the same as the closeness function in bilateral filtering.

Without causing ambiguity, we can use one function to represent both distance-related and edge-sensing weights:

$$w(x, y, s, t, dx, dy) = w_d(s, t) \cdot w_e(x + s, y + t, dx, dy) \quad (19)$$

If the matrix form is used, weighting to all the inter-pixel chrominance in a window can be represented by diagonal matrices, \mathbf{W} , with the corresponding weights as its diagonal elements. Weighting the captured inter-pixel chrominance is equivalent to multiplying the weighting matrix to \mathbf{GF} .

V. LINEAR COMBINATION FOR CHROMINANCE ESTIMATION

The captured inter-pixel chrominance directions are mostly not the same as the target chrominance directions that we need, but they may still contribute to the target if the direction is not perpendicular to. To make the maximum use of the inter-pixel chrominance, we can apply a linear combination to all the inter-pixel chrominance directions in a window to find the target direction. To this end, we have to answer one question: How much can an inter-pixel chrominance component be used for the target chrominance? Or practically the question can be: How to find the coefficients for the linear combination, which also takes into account the distance- or edge-based weighting as described in Section IV? We use the mathematical equations to find the solution by the matrix inverse.

In a window centered at pixel (x, y) , computing all the inter-pixel chrominance directions is equivalent to $\mathbf{G} \cdot \mathbf{F}$, as explained in Section III. Applying weights to all the components of $\mathbf{G} \cdot \mathbf{F}$ can be formulated as $\mathbf{W} \cdot \mathbf{G} \cdot \mathbf{F}$.

If the linear combination coefficients for the weighted inter-pixel chrominance are represented by an unknown row vector \mathbf{x}_1^T , for the target chrominance component $C_{h1} = \mathbf{h}_1^T \cdot \mathbf{i} = p_r R + p_g G + p_b B$, we can have a system of equations for the target chrominance direction:

$$\mathbf{h}_1^T = [p_r, p_g, p_b] = \mathbf{x}_1^T \cdot \mathbf{W} \cdot \mathbf{G} \cdot \mathbf{F} \quad (20)$$

Generally we have $\text{rank}(\mathbf{F}) = 3$ as there are three primary colors and $\text{rank}(\mathbf{W} \cdot \mathbf{G} \cdot \mathbf{F}) = 2$ as the intrinsic dimension of chrominance is 2. In case that there are only 2 colors of pixels in the window, we have $\text{rank}(\mathbf{F}) = 2$ and $\text{rank}(\mathbf{W} \cdot \mathbf{G} \cdot \mathbf{F}) < 2$, then it is impossible to find two independent chrominance components. To fix this problem, we need a bigger window to include three colors of pixels. If weights are unfortunately 0 at some pixels, which rarely happens, we may also have $\text{rank}(\mathbf{W} \cdot \mathbf{G} \cdot \mathbf{F}) < 2$, and in such a case we can correct it with small nonzero weights.

For the system of matrix equation (20) is under-determined, the solution is not unique. To reduce the estimation error, a solution with the minimum norm would be the best. In this paper, the unknown vector \mathbf{x}_1^T is found by the Moore-Penrose pseudo-inverse of the coefficient matrices for the minimum norm least squares solution:

$$\mathbf{x}_1^T = [p_r, p_g, p_b] \cdot (\mathbf{W} \cdot \mathbf{G} \cdot \mathbf{F})^+ \quad (21)$$

If we concatenate all the weights for all the CFA colors in the window around pixel (x, y) and put into a row vector $\hat{\mathbf{h}}_1^T$, we can re-write Eq. (11) in matrix form as follows:

$$\hat{\mathbf{h}}_1^T = [p_r, p_g, p_b] = \hat{\mathbf{h}}_1^T \cdot \mathbf{F} \quad (22)$$

Comparing Eq.(20) and Eq.(22), we can find the weights $\hat{\mathbf{h}}_1^T$ by multiplying all together the matrices of the coefficients \mathbf{x}_1^T ,

the weights \mathbf{W} and the matrix \mathbf{G} for inter-pixel chrominance :

$$\hat{\mathbf{h}}_1^T = \mathbf{x}_1^T \cdot \mathbf{W} \cdot \mathbf{G} = [p_r, p_g, p_b] \cdot (\mathbf{W} \cdot \mathbf{G} \cdot \mathbf{F})^+ \cdot \mathbf{W} \cdot \mathbf{G} \quad (23)$$

Similarly, we can find the weights in the second target chrominance direction $C_{h2} = \mathbf{h}_2^T \cdot \mathbf{i} = q_r R + q_g G + q_b B$:

$$\hat{\mathbf{h}}_2^T = \mathbf{x}_2^T \cdot \mathbf{W} \cdot \mathbf{G} = [q_r, q_g, q_b] \cdot (\mathbf{W} \cdot \mathbf{G} \cdot \mathbf{F})^+ \cdot \mathbf{W} \cdot \mathbf{G} \quad (24)$$

These two weight vectors for the two target chrominance directions can also be put into one matrix equation and found simultaneously:

$$\begin{bmatrix} \hat{\mathbf{h}}_1^T \\ \hat{\mathbf{h}}_2^T \end{bmatrix} = \begin{bmatrix} p_r & p_g & p_b \\ q_r & q_g & q_b \end{bmatrix} \cdot (\mathbf{W} \cdot \mathbf{G} \cdot \mathbf{F})^+ \cdot \mathbf{W} \cdot \mathbf{G} \quad (25)$$

Relating Eq.(25) to Eqs.(8,9,11), we can see that, as a result, $\hat{\mathbf{h}}_1^T$ and $\hat{\mathbf{h}}_2^T$ are the vector version of the weights for chrominance estimation at pixel (x, y) in the window, $\hat{h}_1(x, y, s, t)$ and $\hat{h}_2(x, y, s, t)$, $(s, t) \in \Omega$. By weighted sum over the CFA-filtered image with such chrominance weights, as given in Eq. (9), we can obtain two estimated chrominance components at pixel (x, y) , C_{h1} and C_{h2} , which are the two chrominance amplitudes in the two target chrominance directions at the pixel of interest:

$$\begin{cases} C_{h1}(x, y) = \sum_{s,t} \hat{h}_1(x, y, s, t) \cdot C_f(x + s, y + t) \\ C_{h2}(x, y) = \sum_{s,t} \hat{h}_2(x, y, s, t) \cdot C_f(x + s, y + t) \end{cases} \quad (26)$$

VI. DEMOSAICKING TRANSFORMATION

At each pixel, having estimated two chrominance components, C_{h1} and C_{h2} , together with the acquired image color value, C_f , we can use the optimal demosaicking transform obtained in Step 1 to find the three primary colors, as shown in Eq. (4). This is Step 6 in Phase 3 of our method.

In the following we first present Step 1 in Phase 1: to find the optimal demosaicking transformation matrix \mathbf{D} for a given CFA color and its inverse $\mathbf{T} = \mathbf{D}^{-1}$ as the forward multiplexing matrix, with which we can have the corresponding optimal target chrominance directions for the given CFA color and then use for the estimation of two independent chrominance components.

Similar to that in [11], the optimal demosaicking transform matrix should have the least norm, $\mathbf{D}^* = \arg \min \|\mathbf{D}\|$. If the mean square error is minimized for demosaicking, then the 2-norm of \mathbf{D} should be minimized.

As in Section II-A, we assume the coefficients, r, g, b, R, G, B , are all in the range of $[0,1]$. Without loss of generality, we suppose all the elements of the chrominance direction vectors are in $[-1,1]$. If we have $r \geq g \geq b$, we can find the optimal \mathbf{D} for the minimum 2-norm:

$$\mathbf{D}^* = \begin{bmatrix} 1 & \frac{g}{r+g} & b \\ 1 & -\frac{r}{r+g} & b \\ 1 & 0 & b-1 \end{bmatrix} = \begin{bmatrix} r & g & b \\ 1 & -1 & 0 \\ r+g & r+g & -1 \end{bmatrix}^{-1} \quad (27)$$

Similar to that in [11], we can find the optimal matrices for other cases of CFA colors (r, g, b) by a column permutation in the multiplexing matrix \mathbf{T} , so Eq. (5) and Eq. (27) do not lose generality.

In Table I, we list the optimal demosaicking transform

TABLE I
THE OPTIMAL DEMOSAICKING MATRICES AND THE CORRESPONDING MULTIPLEXING MATRICES FOR DIFFERENT CFA COLORS

CFA color	Optimal demosaicking matrix (\mathbf{D}^*)	Multiplexing matrix ($\mathbf{T} = \mathbf{D}^{-1}$)
Red (R)	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}$
Green (G)	$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & -1 & 1 \end{bmatrix}$
Blue (B)	$\begin{bmatrix} 1 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 0 & -1 \end{bmatrix}$
Cyan (C)	$\begin{bmatrix} 1 & 0 & 1 \\ 1 & -\frac{1}{2} & 0 \\ 1 & \frac{1}{2} & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & -1 & 1 \\ 1 & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix}$
Magenta (M)	$\begin{bmatrix} 1 & \frac{1}{2} & 0 \\ 1 & 0 & 1 \\ 1 & -\frac{1}{2} & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ 1 & 0 & -1 \\ -\frac{1}{2} & 1 & -\frac{1}{2} \end{bmatrix}$
Yellow (Y)	$\begin{bmatrix} 1 & \frac{1}{2} & 0 \\ 1 & -\frac{1}{2} & 0 \\ 1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 1 & -1 & 0 \\ -\frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix}$
White (W)	$\begin{bmatrix} 1 & \frac{1}{2} & -\frac{1}{3} \\ 1 & 0 & \frac{2}{3} \\ 1 & -\frac{1}{2} & -\frac{1}{3} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 1 & 0 & -1 \\ -\frac{1}{2} & 1 & -\frac{1}{2} \end{bmatrix}$
White (W)	$\begin{bmatrix} 1 & 0 & \frac{2}{3} \\ 1 & -\frac{1}{2} & -\frac{1}{3} \\ 1 & \frac{1}{2} & -\frac{1}{3} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & -1 & 1 \\ 1 & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix}$
White (W)	$\begin{bmatrix} 1 & \frac{1}{2} & -\frac{1}{3} \\ 1 & -\frac{1}{2} & -\frac{1}{3} \\ 1 & 0 & \frac{2}{3} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 1 & -1 & 0 \\ -\frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix}$

matrices and the corresponding multiplexing matrices for CFA colors, red (R), green (G), blue (B), cyan (C), magenta (M), yellow (Y) and panchromatic or white (W), among which all the three matrices for CFA white pixels are equally optimal in terms of 2-norm but which is better in practice depends on the CFA pattern.

VII. EXPERIMENTS

For the experimental evaluation of our method, we use the Kodak dataset (Kodak lossless true color image suite, <http://r0k.us/graphics/kodak/>) and the IMAX dataset [33,45] for CFA imaging and demosaicking comparison. The Kodak dataset contains 24 color images of size 768×512, and IMAX has 18 500×500 color images cropped from 8 2310×1814 high-resolution images. The latter looks less smooth and more saturated, and has lower spectral correlation [40,42]. The tested CFA patterns are Bayer CFA [3], the Kodak's CFA2.0 [15], Fuji X-Trans [30], Sony RGBW [31], a random CFA [37] and a

TABLE II
DEMOAICKING PERFORMANCE COMPARISON IN COLOR PSNR

DATASET	CFA	CONDAT'S GENERIC METHOD [6]	MENON & CALVAGNO'S RAD [20]	OURS (ACUDE)
Kodak	Bayer CFA	38.60	39.95	40.84
	Fuji X-Trans	38.25	38.83	39.54
	Random CFA	39.19	39.72	40.10
	Kodak CFA2.0	37.29	37.88	38.70
	Sony RGBW	36.59	37.19	38.10
	Our RGBW	37.88	38.44	38.93
IMAX	Bayer CFA	35.28	36.05	36.38
	Fuji X-Trans	34.85	35.39	35.99
	Random CFA	34.36	35.99	36.25
	Kodak CFA2.0	34.19	34.61	35.15
	Sony RGBW	33.90	34.46	34.87
	Our RGBW	35.02	35.53	36.21

TABLE III
DEMOAICKING PERFORMANCE COMPARISON IN CIE LAB ERROR

DATASET	CFA	CONDAT'S GENERIC METHOD [6]	MENON & CALVAGNO'S RAD [20]	OURS (ACUDE)
Kodak	Bayer CFA	1.515	1.320	1.186
	Fuji X-Trans	1.714	1.569	1.437
	Random CFA	1.527	1.413	1.341
	Kodak CFA2.0	2.146	1.949	1.764
	Sony RGBW	2.279	2.056	1.861
	Our RGBW	1.925	1.780	1.652
IMAX	Bayer CFA	3.621	3.269	3.150
	Fuji X-Trans	3.869	3.558	3.276
	Random CFA	4.036	3.373	3.198
	Kodak CFA2.0	4.764	4.404	3.868
	Sony RGBW	4.827	4.427	3.976
	Our RGBW	4.222	3.935	3.467

5x5 RGBW CFA proposed by us previously [25], as shown in Fig. 3. The other universal demosaicking methods compared to our proposed ACUDE are Condat's generic variational approach [6] and Menon & Calvagno's regularized approach (RAD) [20].

We test the performance in two cases: noise-free and CFA images corrupted with additive Gaussian white noise of standard deviation 5. We measure color PSNR and CIE LAB error between the demosaicked images and the original images for objective comparison, and the results are intercomparable with those published in other publications with the same dataset.

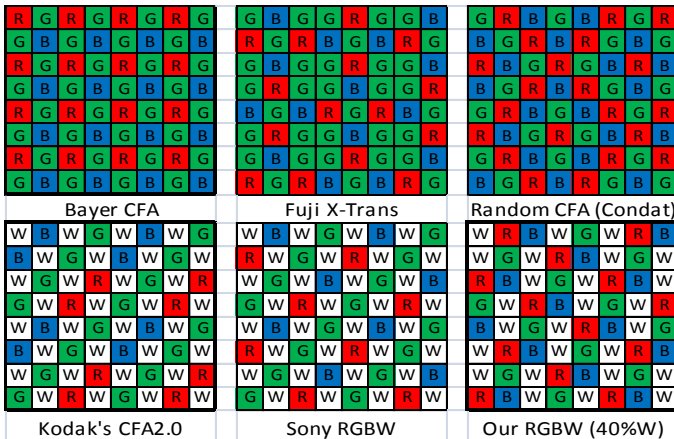


Fig. 3. The tested CFA patterns

TABLE IV
DEMOAICKING PERFORMANCE COMPARISON IN COLOR PSNR
WITH ADDITIVE NOISE OF STANDARD DEVIATION 5

DATASET	CFA	CONDAT'S GENERIC METHOD [6]	MENON & CALVAGNO'S RAD [20]	OURS (ACUDE)
Kodak	Bayer CFA	33.09	33.44	33.49
	Fuji X-Trans	32.86	33.14	33.11
	Random CFA	33.19	33.43	33.38
	Kodak CFA2.0	32.23	32.20	32.52
	Sony RGBW	32.08	32.14	32.49
	Our RGBW	32.65	32.93	32.86
IMAX	Bayer CFA	32.18	32.43	32.41
	Fuji X-Trans	31.83	32.07	32.26
	Random CFA	31.73	32.47	32.60
	Kodak CFA2.0	30.55	30.73	31.04
	Sony RGBW	30.59	30.79	31.14
	Our RGBW	31.42	31.58	32.06

TABLE V
DEMOAICKING PERFORMANCE COMPARISON IN CIE LAB ERROR
WITH ADDITIVE NOISE OF STANDARD DEVIATION 5

DATASET	CFA	CONDAT'S GENERIC METHOD [6]	MENON & CALVAGNO'S RAD [20]	OURS (ACUDE)
Kodak	Bayer CFA	2.767	2.695	2.513
	Fuji X-Trans	2.894	2.807	2.628
	Random CFA	2.766	2.696	2.571
	Kodak CFA2.0	3.364	3.552	3.093
	Sony RGBW	3.365	3.479	3.007
	Our RGBW	3.066	3.012	2.849
IMAX	Bayer CFA	6.720	6.072	5.811
	Fuji X-Trans	6.891	6.247	5.729
	Random CFA	6.926	6.125	5.654
	Kodak CFA2.0	8.859	7.733	6.971
	Sony RGBW	8.663	7.661	6.688
	Our RGBW	7.722	7.004	6.115

As for our experience, the window size for the weighting can be 9-13 pixels in diameter, and for rectangular uniform CFA patterns, the size can be from 9x9 to 13x13. For a window of $(2n + 1) \times (2n + 1)$, n pixels from the image border are excluded from calculation of the demosaicking error.

For the solution of Condat's method is unique [6], the method always converges to the same point no matter what the initial estimate is. So we use our non-adaptive demosaicking estimate for the initial input, which makes the convergence much faster, and it is easier for us to find the optimal parameter μ used in the method, with which the performance is better than those published. Menon & Calvagno's RAD [20] only works for periodic CFAs, thus we only used the first 36x36 sub-pattern of the random CFA for the experiments. Compared to the performance in [20], our results with RAD are a little better because we used a bigger window size to match our experimental configuration for fair comparison.

The averages of the demosaicking results are listed in Table II and III for noise-free and Table IV and V for with noise. For space limitations, we only present the results of our noise-free experiments for individual test images in Table VI to IX. Please use the link given at the end of this section for more details about all our experiments.

The tables for the noise-free experiments with Kodak dataset show that in average our proposed method outperforms Condat's method by 1.4dB in color PSNR (or 3.34 in color MSE) and by 0.31 in CIE LAB error and outperforms Menon &

Calvagno's RAD by 0.7dB in color PSNR (or 1.60 in color MSE) and 0.14 in CIE LAB error, respectively. With IMAX dataset, the performance is similar: Ours does averagely 1.2dB higher than Condat's and 0.47dB better than RAD in CPSNR (or CMSE 6.63 and 2.63 less), and CIE LAB error average is 0.73 and 0.34 better than Condat's and RAD, respectively. We can see that our new demosaicking method performs the best in average and the best for most of the test images in both datasets among the three demosaicking methods.

If we compare the results with those state-of-the-art methods presented in other publications for demosaicking Bayer CFA, e.g. [18,22], we can see that our demosaicking performance is even better than many demosaicking methods that are specially designed for Bayer CFA. For this purpose, we also include the results with Kodak dataset in [22] into Table VI and VII. It shows that our method performs very similar to that in [22], and better than those methods compared therein.

Parts of image 19 (lighthouse) in the Kodak dataset are given in Fig. 4. For the figure, the original is shown at the top, and the other 6 rows are for the 6 CFAs and the 3 columns for the 3 universal methods. The images show that our proposed method gives the least false color and zipper effect.

The experiments with additive noise, as the averages shown in Table IV and V, tell that in terms of CPSNR our method performs better than Condat's and close to RAD for both datasets, but in terms of LAB error, ours is unanimously the best.

In addition, if we compare the performance between CFAs, we can see that among all the six tested CFAs the Bayer CFA objectively does the best for noise-free experiments, but the random CFA does the best subjectively and in most of the noisy cases. For the CFAs with 40% or above white pixels, Kodak CFA2.0, Sony RGBW and our RGBW patterns, ours does the best and the demosaicked image quality is even comparable to the random CFA for all the three demosaicking methods.

The computational complexity of our method in the demosaicking phase is equivalent to two image filtering processes for the two chrominance components plus a 3D color space transformation. In the weight generation phase, the weights can be generated off-line, but the adaptive weights can only be done on-line and it involves a matrix pseudo-inverse at each pixel, which is not so efficient. However, the overall processing speed of our method is the same for all CFAs.

More details of the experimental results and all the images demosaicked with all the three demosaicking methods from CFA images simulated with all the six CFA patterns presented in this paper and a few other CFAs are available on web at <http://www.eecs.qmul.ac.uk/~phao/CFA/acude/>.

VIII. CONCLUSION

In this paper, we proposed an adaptive chrominance-based universal demosaicking method (ACUDe), which outperforms two other universal demosaicking methods, and its performance with Bayer CFA is even better than many demosaicking methods that are specially designed for Bayer CFA. ACUDe reduces demosaicking error by avoiding the

commonly used step to estimate a luminance component at a pixel. The main error of ACUDe comes from the target chrominance estimation, but the error can be further reduced by using an edge-sensing weighting mechanism and the optimal demosaicking transformation, which minimize the error transferred from the estimated target chrominance to the primary color space.

The regularization approaches [20] minimize the error between the acquired CFA image and the simulated CFA image from the demosaicked, so it can better reduce the noise mixed into the CFA image. With ACUDe and Condat's generic method [6], this mixed noise is not reduced but retained: the demosaicked image gives a simulated CFA image exactly the same as the acquired CFA image. In future, further research is needed to make our method more robust to noise.

The implementation of the proposed method can be faster, especially with some specialized hardware for fast matrix pseudo-inverse. This is also worth further investigation.

Although the proposed method is for the three primary color system, the demosaicking idea, the universal method and the theory can all be applied to multispectral imaging with multispectral filter arrays [47] by using a different primary color system.

ACKNOWLEDGMENT

The authors would like to thank Dr Laurent Condat, Dr Menon and Dr Calvagno for providing their codes for our experiments, and also wish to thank all the anonymous reviewers for their constructive comments, with which we could make the paper more complete and clearer.

REFERENCES

- [1] J.E. Adams, and J.F. Hamilton, "Adaptive color plane interpolation in single sensor color electronic camera", US Patent 5 652 621, 1997.
- [2] D. Alleysson, S. Susstrunk, and J. Herault, "Linear demosaicing inspired by the human visual system", *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 439-449, April 2005.
- [3] B. E. Bayer, "Color imaging array", US Patent 3 971 065, 1976.
- [4] A. Ben-Israel, and T.N.E. Greville, "*Generalized Inverses: Theory and Applications*", CMS Books in Mathematics, Springer, 2003.
- [5] K.-H. Chung, and Y.-H. Chan, "Color demosaicing using variance of color differences", *IEEE Transactions on Image Processing*, vol. 15, no. 10, p. 2944-2955, October 2006.
- [6] L. Condat, "A generic variational approach for demosaicking from an arbitrary color filter array", *IEEE International Conference on Image Processing (ICIP)*, Cairo, Egypt, Nov 7-10, 2009, pp. 1625-8.
- [7] E. Dubois, "Frequency-domain methods for demosaicking of Bayer sampled colour images", *IEEE Signal Processing Letters*, vol. 12, no. 12, pp. 847-850, 2005.
- [8] J. Gu, P. J. Wolfe, and K. Hirakawa, "Filterbank-based universal demosaicking", *IEEE International Conference on Image Processing (ICIP)*, Hong Kong, Sept 26-29, 2010, pp.1981-4.
- [9] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections", *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 997-1013, September 2002.
- [10] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosaicking: color filter array interpolation", *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 44-54, January 2005.
- [11] P. Hao, Y. Li, Z. Lin, and E. Dubois, "A geometric method for optimal design of color filter arrays", *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 709-722, March 2011.

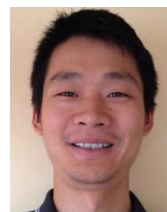
- [12] J.F. Hamilton, J.E. Adams, and D.M. Orlicki, "Particular pattern of pixels for a color filter array which is used to derive luminance and chrominance values", US Patent 6 330 029 B1, Dec. 2001.
- [13] K. Hirakawa, and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm", *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 360-369, March 2005.
- [14] A. Hore, and D. Ziou, "An Edge-Sensing Generic Demosaicing Algorithm with Application to Image Resampling", *IEEE Transactions on Image Processing*, vol. 20, no. 11, pp. 3136-3150, November 2011.
- [15] T. Kijima, H. Nakamura, J. Compton, and J. Hamilton, "Image sensor with improved light sensitivity", US Patent 20070268533, 22 Nov 2007.
- [16] Y. Li, P. Hao, and Z. Lin, "The Frequency Structure Matrix: A Representation of Color Filter Arrays", *International Journal of Imaging Systems and Technology*, vol. 21, no. 1, pp. 101-106, 2011.
- [17] Y. Li, P. Hao, and Z. Lin, "Color Filter Arrays: Representation and Analysis", *Research Report, Dept of Computer Science, Queen Mary, University of London*, RR-08-04, ISSN 1470-5559, May 2008.
- [18] N.-X. Lian, L. Chang, Y.-P. Tan, and V. Zagorodnov, "Adaptive Filtering for Color Filter Array Demosaicking", *IEEE Transactions on Image Processing*, vol. 16, no. 10, pp. 2515-2525, October 2007.
- [19] R. Lukac, K.N. Plataniotis, "Universal demosaicking for imaging pipelines with a RGB color filter array", *Pattern Recognition*, vol. 38, pp. 2208-2212, 2005.
- [20] D. Menon, and G. Calvagno, "Regularization approaches to demosaicking", *IEEE Transactions on Image Processing*, vol. 18, no. 10, pp. 2209-2210, October 2009.
- [21] D. Menon, and G. Calvagno, "Color image demosaicking: An overview", *Signal Processing: Image Communication*, vol. 26, no. 8-9, pp. 518-533, October 2011.
- [22] I. Pekkucuksen, and Y. Altunbasak, "Multiscale Gradients-Based Color Filter Array Interpolation", *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 157-165, Jan. 2013.
- [23] M. Singh, and T. Singh, "Linear universal demosaicking of regular pattern color filter arrays", *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, March 25-30, 2012, pp. 1277-80.
- [24] H.J. Trussell and R.E. Hartwig, "Mathematics for demosaicking", *IEEE Transactions on Image Processing*, vol. 3, no. 11, pp. 485-492, April 2002.
- [25] J. Wang, C. Zhang, and P. Hao, "New Color Filter Arrays of High Light Sensitivity and High Demosaicking Performance", *IEEE International Conference on Image Processing (ICIP)*, Brussels, Belgium, Sept 11-14, 2011, pp. 3153-6.
- [26] M. Wang, and T. Blu, "Generalized YUV interpolation of CFA images", *International Conference on Image Processing (ICIP)*, Hong Kong, Sept 26-29, 2010, pp. 1909-12.
- [27] L. Zhang, and X. Wu, "Color Demosaicking via directional linear minimum mean square-error estimation", *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2167-2178, December 2005.
- [28] M. Kumar, E. Morales, J. Adams, and W. Hao, "New digital camera sensor architecture for low light imaging", *IEEE International Conference on Image Processing (ICIP)*, Cairo, Egypt, Nov 7-10, 2009, pp. 2681-4.
- [29] [Online] Digital Photography Review, "New honeycomb "Super CCD" from FujiFilm", DPReview, 20 October 1999. Available: <http://www.dpreview.com/news/9910/99102003superccd.asp>
- [30] [Online] Fujifilm X-Pro1, "X-Trans CMOS", 2012, available: <http://fujifilm-x.com/x-pro1/en/about/sensor/index.html>
- [31] [Online] Sony Image Sensor Business Presentation, Semiconductor Business Meeting, Aug 30, 2011, available: <http://image-sensors-world.blogspot.co.uk/2011/09/sony-image-sensor-business-presentation.html>
- [32] [Online] Sony Press Release, "Realization of natural color reproduction in digital still cameras, closer to the natural sight perception of the human eye", 16 July 2003. Available: <http://www.sony.net/SonyInfo/News/Press/200307/03-029E/>.
- [33] X. Li, B. Gunturk, and L. Zhang, "Image demosaicing: A systematic survey", *Proceedings of SPIE*, vol. 6822, pp. 68221J:1-15, January 2008.
- [34] O. Losson, and E. Dinet, "From the sensor to color images", in *Digital Color: Acquisition, Perception, Coding and Rendering*, C. Fernandez-Maloigne, F. Robert-Inacio, and L. Macaire (ed.), Wiley, 2012, pp.149-185.
- [35] K. Hirakawa, and P.J. Wolfe, "Spatio-spectral color filter array design for optimal image recovery", *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1876-1890, October 2008.
- [36] L. Condat, "A new color filter array with optimal properties for noiseless and noisy color image acquisition", *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2200-2210, August 2011.
- [37] L. Condat, "Color filter array design using random patterns with blue noise chromatic spectra", *Image and Vision Computing*, vol. 28, no. 8, pp. 1196-1202, August 2010.
- [38] B. Leung, G. Jeon, and E. Dubois, "Least-squares luma-chroma demultiplexing algorithm for Bayer demosaicking", *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1885-1894, July 2011.
- [39] S. Ferradans, M. Bertalmio, and V. Caselles, "Geometry-based demosaicking", *IEEE Transactions on Image Processing*, vol. 18, no. 3, pp.665-670, March 2009.
- [40] A. Buades, B. Coll, J.-M. Morel, and C. Sbert, "Self-similarity driven color demosaicking", *IEEE Transactions on Image Processing*, vol. 18, no. 6, pp. 1192-1202, June 2009.
- [41] 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 Transactions on Image Processing*, vol. 18, no. 9, pp. 1946-1957, Sept 2009.
- [42] F. Zhang, X. Wu, X. Yang, W. Zhang, and D. Zhang, "Robust color demosaicking with adaptation to varying spectral correlations", *IEEE Transactions on Image Processing*, vol. 18, no. 12, pp. 2706-2717, December 2009.
- [43] C. Kim, H. Oh, D. Yoo, and M. Kang, "Region adaptive color demosaicking algorithm using color constancy", *EURASIP Journal on Advances in Signal Processing*, vol. 2010, pp. 1-18, April 2010.
- [44] K.-H. Chung, and Y.-H. Chan, "Low-complexity color demosaicking algorithm based on integrated gradients", *Journal of Electronic Imaging*, vol. 19, no. 2, pp. 021104:1-15, June 2010.
- [45] L. Zhang, X. Wu, A. Buades, and X. Li, "Color demosaicking by local directional interpolation and non-local adaptive thresholding", *Journal of Electronic Imaging*, vol. 20, no. 2, pp. 023016:1-16, April 2011.
- [46] S.W. Park, and M.G. Kang, "Channel correlated refinement for color interpolation with quincuncial patterns containing the white channel", *Digital Signal Processing*, vol. 23, no. 5, pp. 1363-1389, September 2013.
- [47] L. Miao, and H. Qi, "The design and evaluation of a generic method for generating mosaicked multispectral filter arrays", *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2780-2791, September 2006
- [48] T. Singh, and M. Singh, "Disregarding spectral overlap - A unified approach for demosaicking, compressive sensing and color filter array design", *IEEE International Conference on Image Processing (ICIP)*, Brussels, Belgium, Sept 11-14, 2011, pp.3161-4.
- [49] C. Tomasi, and R. Manduchi, "Bilateral filtering for gray and color images", *International Conference on Computer Vision (ICCV)*, Bombay, India, Jan 4-7, 1998, pp. 839-846.
- [50] L. Condat, "A Generic Proximal Algorithm for Convex Optimization - Application to Total Variation Minimization", *IEEE Signal Proc. Letters*, vol. 21, no. 8, pp. 1054-1057, August 2014.



Chao Zhang (M'06) received the Ph.D. degree in electrical engineering from Beijing Jiaotong University, Beijing, China, in 1995.

He was a Post-Doctoral Research Fellow with the National Laboratory on Machine Perception, Peking University, Beijing, from 1995 to 1997. He has been an Associate Professor with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University, since 1997. His current research interests include image processing,

statistical pattern recognition, and visual recognition.



Yan Li received the B.Sc. degree from Nanchang University in 2004 and the M.Sc. degree from Peking University in 2007, both in electrical engineering. He received the Ph.D. degree from Delft University of Technology in 2013. His Ph.D. dissertation is on multi-scale pattern recognition with applications in image segmentation and classification. Since then he has been working in Lely Technologies as a R&D researcher, developing autonomous agriculture robots.



Jue Wang received the B.Sc. degree in computer science from Jilin University in 2009 and the M.Sc. degree in intelligent science from Peking University in 2012, and currently is a PhD candidate in software engineering at Peking University. Her research interests are on algorithms of color imaging, demosaicing and video analysis, such as intrusion detection and crowd density estimation, which have been applied in intelligent video monitoring system for modern

railways in China.



Pengwei Hao (M'98) received the B.Sc. degree in computer science and the M.Sc. degree in computer graphics from Northwestern Polytechnical University, Xi'An, China, in 1988 and 1994, respectively. He received the Ph.D. degree in image processing from the Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing, China, in 1997.

From 1997 to 1999 he was a Lecturer and was promoted to an Associate Professor in 1999 with the Center for Information Science, Peking University, Beijing, China. Since 2002, he has been a Lecturer at Queen Mary, University of London, UK. He was a Visiting Scientist with the Centre for Vision, Speech, and Signal Processing, University of Surrey, UK, in 2000, a Visiting Researcher with the School of Information Technology and Engineering (SITE), University of Ottawa, Canada, in 2008 and a Courtesy Associate Professor with the Department of Electrical & Computer Engineering, University of Florida, USA, in 2009. His research interests include data and image coding, signal sampling and reconstruction, integer transforms, and color imaging.

TABLE VI
NOISE-FREE DEMOSAICKING PERFORMANCE COMPARISON IN COLOR PSNR WITH KODAK DATASET

CFA pattern: Demosaic image	BAYER CFA				FUJI X-TRANS			RANDOM CFA			KODAK CFA2.0			SONY RGBW			OUR RGBW		
	[22]	[6]	[20]	OURS	[6]	[20]	OURS	[6]	[20]	OURS	[6]	[20]	OURS	[6]	[20]	OURS	[6]	[20]	OURS
1	39.87	37.02	38.32	39.56	37.50	37.21	38.42	38.75	38.56	39.33	36.73	36.99	38.09	35.57	34.95	36.98	37.18	36.94	37.96
2	41.77	39.04	39.99	41.48	38.83	39.26	40.24	39.66	39.91	40.69	38.13	38.35	39.55	37.25	37.95	38.80	38.59	38.97	39.69
3	43.72	40.77	42.63	42.99	39.89	41.22	41.65	40.72	41.86	42.24	39.20	39.83	41.10	38.12	40.12	40.10	38.66	40.06	40.34
4	41.13	40.15	41.14	41.24	39.27	39.99	39.85	40.20	40.77	40.58	38.07	38.61	38.71	37.22	38.40	37.82	38.71	39.52	39.20
5	39.05	36.94	38.44	38.89	35.35	36.42	36.75	36.35	37.39	37.25	33.71	34.82	35.24	32.90	34.75	34.56	34.83	35.93	36.07
6	41.38	38.00	39.86	41.28	38.35	38.72	39.97	39.60	40.00	40.59	37.61	38.18	38.96	37.01	36.46	38.56	38.47	38.59	39.37
7	43.51	41.33	43.04	43.51	40.36	41.31	41.76	41.31	42.07	42.33	38.75	39.40	40.31	38.06	39.52	39.71	39.69	40.73	40.98
8	37.56	33.03	35.93	37.83	35.23	35.38	37.38	36.62	36.83	38.22	34.39	34.89	37.05	33.43	31.85	36.62	35.39	35.49	37.12
9	43.96	40.89	42.58	43.82	40.77	41.40	42.49	41.69	42.22	43.02	39.57	40.11	41.49	38.98	39.58	41.11	40.41	41.12	41.96
10	43.20	41.70	42.67	43.26	40.85	41.54	42.06	41.66	42.39	42.66	39.52	40.19	41.02	39.41	40.23	40.73	39.42	40.57	40.71
11	41.36	38.61	39.84	41.10	38.22	38.63	39.61	39.34	39.66	40.22	37.18	37.72	38.71	36.58	36.90	38.18	38.03	38.31	39.02
12	44.45	41.98	43.64	44.32	42.07	42.54	43.22	43.14	43.60	43.97	41.27	41.51	42.55	40.35	40.49	41.79	41.70	42.30	42.82
13	36.00	34.77	34.61	35.78	33.16	33.18	34.18	34.04	34.12	34.72	32.60	32.87	33.67	32.03	32.35	32.98	33.05	32.91	33.80
14	37.97	34.55	36.50	37.28	34.13	35.38	35.57	35.09	36.16	36.05	32.73	33.81	34.26	32.03	33.80	33.59	33.55	34.80	34.52
15	40.30	39.42	40.04	40.15	38.32	38.92	38.79	39.26	39.77	39.59	37.51	37.92	38.13	36.94	37.76	37.52	37.85	38.37	38.28
16	44.86	40.91	43.40	44.93	41.86	42.15	43.49	43.23	43.46	44.06	41.32	41.65	42.63	40.42	39.47	42.21	41.59	41.66	42.62
17	42.32	41.15	41.62	42.34	39.97	40.46	41.08	40.54	41.12	41.32	39.39	40.04	40.55	38.73	39.83	40.15	39.79	40.32	40.70
18	38.22	36.96	37.29	38.08	35.79	36.17	36.68	36.36	36.71	36.98	34.94	35.49	35.88	34.70	35.51	35.39	35.59	35.94	36.30
19	42.17	37.94	40.16	41.72	39.33	39.61	40.90	40.38	40.62	41.49	38.37	39.05	40.28	37.59	36.76	39.78	39.44	39.54	40.58
20	42.16	40.04	41.08	42.05	39.25	40.01	40.87	40.10	40.78	41.37	38.48	39.15	40.32	37.93	39.08	39.80	38.68	39.43	40.32
21	40.31	38.21	39.25	40.33	38.14	38.33	39.29	39.05	39.31	39.80	36.81	37.58	38.18	36.38	36.32	37.81	38.06	38.20	38.85
22	39.05	37.26	38.46	39.03	37.02	37.75	37.89	37.59	38.28	38.36	36.06	36.79	37.08	35.52	36.31	36.59	36.89	37.59	37.58
23	44.02	41.16	42.89	43.65	40.60	41.71	42.02	41.60	42.45	42.38	38.92	39.74	40.45	38.13	40.01	39.70	39.99	41.20	41.15
24	35.69	34.63	35.32	35.62	33.82	34.57	34.88	34.32	35.18	35.13	33.58	34.38	34.51	32.97	34.13	33.84	33.50	34.13	34.50
Avg.	41.00	38.60	39.95	40.84	38.25	38.83	39.54	39.19	39.72	40.10	37.29	37.88	38.70	36.59	37.19	38.10	37.88	38.44	38.93

TABLE VII
NOISE-FREE DEMOSAICKING PERFORMANCE COMPARISON IN CIE LAB ERROR WITH KODAK DATASET

CFA pattern: Demosaic image	BAYER CFA				FUJI X-TRANS			RANDOM CFA			KODAK CFA2.0			SONY RGBW			OUR RGBW		
	[22]	[6]	[20]	OURS	[6]	[20]	OURS	[6]	[20]	OURS	[6]	[20]	OURS	[6]	[20]	OURS	[6]	[20]	OURS
1	1.274	1.750	1.561	1.325	1.857	1.881	1.632	1.612	1.622	1.480	2.247	2.145	1.895	2.476	2.572	2.075	2.118	2.147	1.921
2	1.323	1.957	1.663	1.374	2.035	1.814	1.599	1.837	1.689	1.547	2.442	2.200	1.876	2.720	2.278	2.037	2.245	2.018	1.826
3	0.782	1.125	0.870	0.829	1.269	1.001	0.945	1.127	0.923	0.885	1.472	1.271	1.094	1.607	1.210	1.161	1.511	1.209	1.160
4	1.096	1.209	1.136	1.096	1.410	1.334	1.324	1.338	1.286	1.289	1.823	1.739	1.686	1.951	1.772	1.829	1.677	1.583	1.589
5	1.729	2.406	1.939	1.784	3.061	2.550	2.351	2.687	2.269	2.218	4.178	3.490	3.167	4.527	3.537	3.367	3.500	2.953	2.739
6	0.931	1.305	1.107	0.945	1.405	1.340	1.168	1.203	1.151	1.067	1.725	1.578	1.477	1.805	1.817	1.532	1.555	1.532	1.393
7	0.907	1.190	0.975	0.913	1.364	1.189	1.094	1.202	1.073	1.000	1.778	1.653	1.384	1.916	1.613	1.452	1.545	1.356	1.232
8	1.393	2.118	1.728	1.379	2.039	2.009	1.593	1.768	1.736	1.471	2.484	2.404	1.867	2.639	2.958	1.924	2.238	2.222	1.835
9	0.689	0.926	0.776	0.697	1.006	0.906	0.810	0.931	0.850	0.777	1.245	1.152	0.989	1.297	1.196	1.016	1.125	1.011	0.920
10	0.724	0.816	0.777	0.724	0.933	0.902	0.837	0.880	0.848	0.801	1.203	1.162	1.034	1.227	1.172	1.062	1.089	1.038	0.965
11	1.132	1.607	1.382	1.180	1.752	1.640	1.428	1.520	1.446	1.321	2.184	2.016	1.750	2.294	2.211	1.817	1.916	1.850	1.634
12	0.558	0.683	0.615	0.569	0.737	0.716	0.661	0.672	0.651	0.619	0.865	0.873	0.775	0.939	0.930	0.827	0.846	0.814	0.757
13	2.177	2.556	2.542	2.234	3.277	3.156	2.868	2.844	2.752	2.583	4.001	3.687	3.430	4.187	3.881	3.666	3.584	3.541	3.227
14	1.501	2.703	1.832	1.607	2.966	2.147	1.985	2.560	1.919	1.858	3.805	2.785	2.569	4.082	2.850	2.729	3.345	2.501	2.396
15	1.248	1.446	1.350	1.290	1.659	1.554	1.521	1.592	1.507	1.486	2.127	2.035	1.903	2.226	2.019	1.998	1.936	1.816	1.777
16	0.789	1.092	0.907	0.789	1.106	1.072	0.946	0.960	0.939	0.881	1.300	1.249	1.150	1.372	1.465	1.186	1.270	1.272	1.153
17	1.206	1.348	1.297	1.203	1.588	1.510	1.415	1.482	1.414	1.364	1.897	1.792	1.672	1.960	1.811	1.705	1.721	1.648	1.559
18	2.020	2.389	2.136	2.033	2.912	2.624	2.560	2.578	2.363	2.355	3.684	3.261	3.165	3.888	3.318	3.349	3.210	2.906	2.839
19	1.027	1.372	1.217	1.045	1.494	1.432	1.274	1.312	1.271	1.174	1.812	1.663	1.507	1.939	1.892	1.588	1.591	1.563	1.413
20	0.868	1.093	0.976	0.883	1.266	1.144	1.035	1.131	1.027	0.952	1.519	1.383	1.203	1.587	1.387	1.254	1.431	1.295	1.158
21	1.121	1.374	1.284	1.126	1.547	1.516	1.362	1.375	1.349	1.273	2.021	1.857	1.728	2.067	1.996	1.776	1.703	1.694	1.566
22	1.305	1.439	1.331	1.299	1.625	1.551	1.553	1.486	1.420	1.426	2.082	1.990	1.906	2.185	2.039	2.006	1.826	1.734	1.732
23	0.826	0.963	0.871	0.843	1.070	0.993	0.981	0.971	0.918	0.923	1.356	1.344	1.207	1.449	1.280	1.281	1.197	1.098	1.095
24	1.274	1.487	1.400	1.289	1.760	1.675	1.554	1.582	1.495	1.444	2.252	2.037	1.907	2.354	2.132	2.031	2.012	1.921	1.765
Avg.	1.162	1.515	1.320	1.186	1.714	1.569	1.437	1.527	1.413	1.341	2.146	1.949	1.764	2.279	2.056	1.861	1.925	1.780	1.652

TABLE VIII
NOISE-FREE DEMOSAICKING PERFORMANCE COMPARISON IN COLOR PSNR WITH IMAX DATASET

CFA pattern: Demosaic image	BAYER CFA			FUJI X-TRANS			RANDOM CFA			KODAK CFA2.0			SONY RGBW			OUR RGBW		
	[6]	[20]	OURS	[6]	[20]	OURS	[6]	[20]	OURS	[6]	[20]	OURS	[6]	[20]	OURS	[6]	[20]	OURS
1	28.33	28.78	29.13	28.01	28.34	28.98	28.15	28.88	29.24	27.46	27.73	28.55	27.34	27.67	28.43	28.14	28.51	29.32
2	33.96	34.60	35.03	33.79	34.09	34.76	33.65	34.66	35.18	33.49	33.79	34.64	33.15	33.60	34.29	33.84	34.16	34.95
3	31.47	32.12	32.76	30.51	31.22	32.34	30.86	31.77	32.77	29.89	30.44	31.07	29.25	30.10	30.73	30.72	31.36	32.38
4	34.46	36.46	37.10	33.61	35.31	37.20	32.96	36.39	37.45	33.40	34.69	35.55	33.07	34.45	35.11	34.80	36.04	37.63
5	33.28	33.82	34.35	33.02	33.38	33.95	32.62	34.07	34.07	32.11	32.33	33.07	32.08	32.41	33.05	33.21	33.56	34.37
6	37.14	38.16	38.12	36.74	37.28	37.13	36.46	38.43	37.42	35.78	35.97	36.09	35.82	36.16	35.77	36.91	37.50	37.39
7	34.68	34.54	35.24	34.48	34.24	34.99	33.90	34.44	35.70	33.24	33.11	33.52	32.54	32.50	33.55	34.36	34.01	35.21
8	36.45	37.12	37.65	35.96	36.39	37.05	36.19	36.94	37.52	35.16	35.73	35.47	34.47	35.29	35.02	36.02	36.45	36.91
9	36.14	37.23	37.79	35.56	36.40	37.05	35.07	37.24	37.25	34.92	35.58	36.18	34.61	35.43	35.74	35.87	36.71	37.49
10	38.07	38.80	38.87	37.37	38.02	38.43	36.81	38.82	38.80	36.97	37.47	37.81	36.69	37.33	37.37	37.66	38.34	38.90
11	38.82	39.32	39.31	38.21	38.63	38.81	37.55	39.12	39.15	37.17	37.45	37.95	37.05	37.39	37.83	38.35	38.80	39.34
12	37.08	38.38	38.68	36.62	37.51	38.49	35.40	37.64	38.33	35.89	36.54	37.52	35.36	36.43	37.23	36.38	37.20	38.17
13	39.64	40.50	40.93	39.68	40.00	40.61	37.27	40.73	40.99	39.27	39.52	40.57	39.10	39.41	40.37	39.99	40.29	41.22
14	37.86	38.31	38.89	37.59	37.91	38.50	36.63	38.48	38.63	37.62	37.87	38.56	37.10	37.69	38.12	37.80	38.21	38.66
15	38.46	38.80	39.19	38.00	38.29	38.74	37.13	38.84	38.90	37.30	37.54	37.98	37.18	37.50	37.78	38.24	38.56	39.09
16	32.65	33.54	33.44	32.22	33.06	33.17	31.82	33.08	32.75	30.95	31.54	31.92	30.83	31.58	31.63	31.78	32.60	32.78
17	33.10	33.79	33.36	32.11	32.65	32.38	32.57	33.69	32.77	31.47	31.87	31.67	31.53	31.97	31.27	32.35	33.06	32.75
18	33.43	34.61	35.06	33.90	34.31	35.22	33.40	34.58	35.49	33.36	33.76	34.64	32.95	33.33	34.44	33.98	34.20	35.27
Avg.	35.28	36.05	36.38	34.85	35.39	35.99	34.36	35.99	36.25	34.19	34.61	35.15	33.90	34.46	34.87	35.02	35.53	36.21

TABLE IX
NOISE-FREE DEMOSAICKING PERFORMANCE COMPARISON IN CIE LAB ERROR WITH IMAX DATASET

CFA pattern: Demosaic image	BAYER CFA			FUJI X-TRANS			RANDOM CFA			KODAK CFA2.0			SONY RGBW			OUR RGBW		
	[6]	[20]	OURS	[6]	[20]	OURS	[6]	[20]	OURS	[6]	[20]	OURS	[6]	[20]	OURS	[6]	[20]	OURS
1	5.803	5.418	5.221	6.232	5.857	5.290	6.027	5.453	5.023	7.473	7.075	6.145	7.521	7.083	6.190	6.574	6.295	5.453
2	4.356	4.151	3.869	4.526	4.412	4.048	4.753	4.342	4.075	5.448	5.252	4.562	5.556	5.326	4.699	5.226	5.090	4.473
3	4.561	3.946	3.624	5.162	4.457	3.752	4.980	4.166	3.604	6.359	5.532	4.657	6.741	5.671	4.875	5.565	4.910	4.018
4	1.689	1.330	1.202	1.778	1.533	1.196	2.250	1.371	1.153	2.082	1.876	1.504	2.134	1.912	1.594	1.743	1.604	1.221
5	2.691	2.547	2.388	2.827	2.745	2.535	3.110	2.612	2.529	3.523	3.439	3.079	3.521	3.411	3.090	3.067	3.017	2.697
6	2.343	2.193	2.198	2.500	2.420	2.383	2.690	2.257	2.324	3.081	3.049	2.830	3.068	3.003	2.919	2.787	2.693	2.548
7	3.030	3.046	2.754	3.173	3.241	2.991	3.462	3.203	2.863	4.080	4.122	3.760	4.276	4.318	3.719	3.484	3.621	3.165
8	4.716	4.425	4.147	4.905	4.696	4.318	4.840	4.553	4.224	5.943	5.599	4.929	6.021	5.659	5.025	5.338	5.153	4.532
9	3.310	2.895	2.604	3.430	3.100	2.775	3.699	2.916	2.775	4.425	3.908	3.290	4.502	3.925	3.426	3.790	3.410	2.918
10	3.152	2.907	2.838	3.349	3.133	2.984	3.712	3.001	2.960	4.144	3.887	3.486	4.169	3.901	3.608	3.690	3.465	3.154
11	3.895	3.457	3.518	4.200	3.786	3.636	4.301	3.434	3.361	5.147	4.696	4.143	5.103	4.652	4.220	4.397	4.022	3.594
12	2.312	2.060	1.947	2.511	2.314	2.036	2.930	2.244	2.020	2.933	2.748	2.297	3.080	2.776	2.371	2.669	2.509	2.148
13	1.471	1.355	1.312	1.463	1.444	1.376	1.907	1.351	1.308	1.645	1.631	1.479	1.671	1.654	1.505	1.544	1.533	1.396
14	2.654	2.579	2.442	2.746	2.710	2.569	3.125	2.660	2.608	3.192	3.162	2.873	3.260	3.190	2.974	3.079	3.036	2.830
15	3.260	3.141	2.884	3.354	3.284	3.081	3.753	3.179	3.128	4.270	4.200	3.629	4.275	4.191	3.690	3.689	3.635	3.274
16	6.352	4.826	5.228	7.127	5.456	5.151	6.383	4.864	4.695	8.616	6.872	6.101	8.593	6.809	6.391	7.398	5.893	5.216
17	5.532	4.968	4.982	6.156	5.544	5.325	6.455	5.427	5.589	8.210	7.483	6.785	8.084	7.330	7.098	7.311	6.568	6.056
18	4.046	3.606	3.548	4.205	3.909	3.515	4.272	3.685	3.320	5.184	4.744	4.081	5.313	4.867	4.170	4.644	4.370	3.706
Avg.	3.621	3.269	3.150	3.869	3.558	3.276	4.036	3.373	3.198	4.764	4.404	3.868	4.827	4.427	3.976	4.222	3.935	3.467

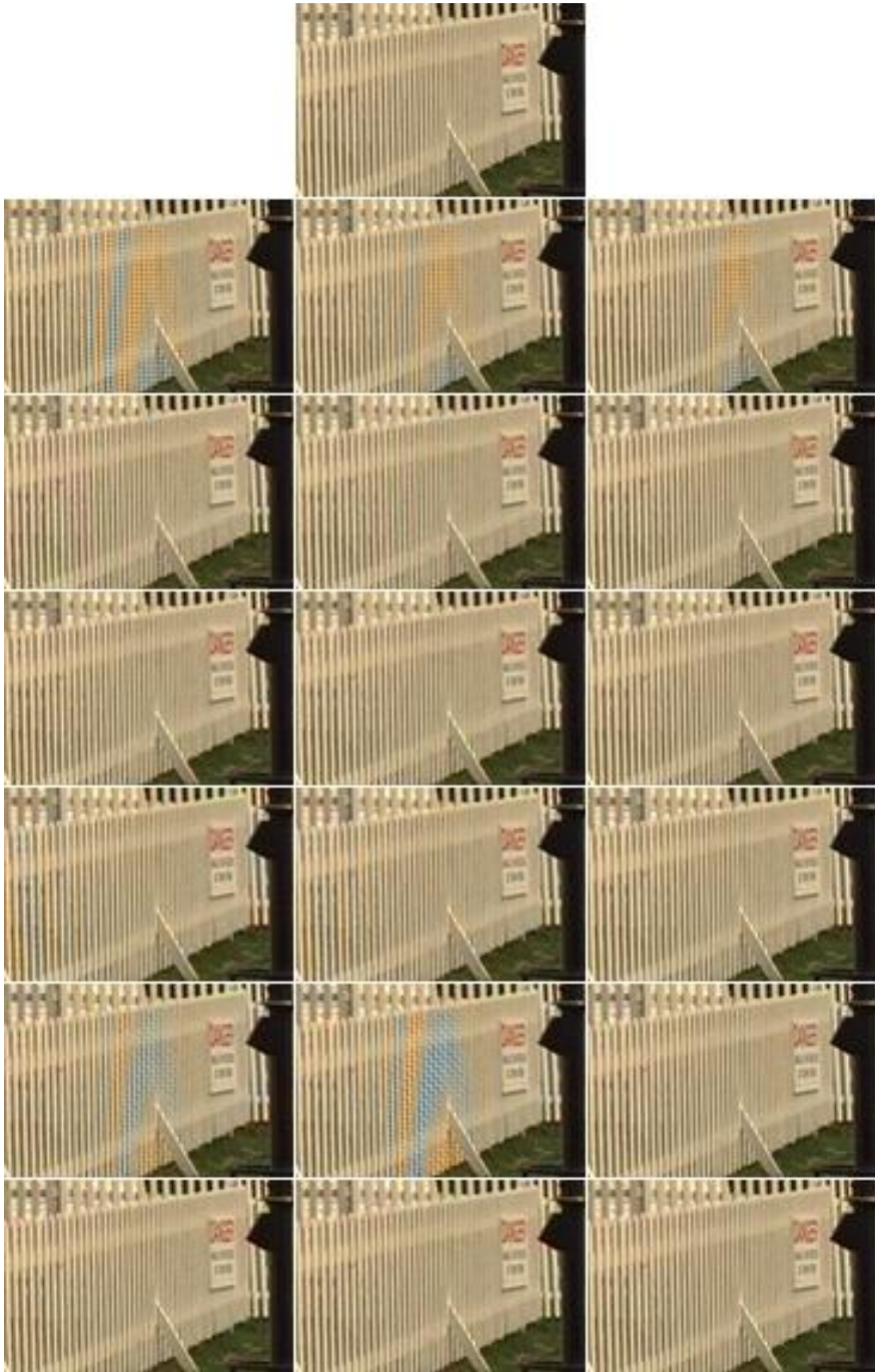


Figure 4. Part of image 19 (top: original) of the Kodak dataset simulated with 6 CFAs (from the second to bottom row: Bayer CFA, Fuji X-Trans, Random CFA, Kodak CFA2.0, Sony RGBW, and our RGBW) and demosaicked with 3 universal methods (from left to right column: Condat's generic variational method [6], Menon & Calvagno's RAD [20], and our ACUDe)