

Demosaicking by Alternating Projections: Theory and Fast One-Step Implementation

Yue M. Lu, *Member, IEEE*, Mina Karzand, and Martin Vetterli, *Fellow, IEEE*

Abstract—Color image demosaicking is a key process in the digital imaging pipeline. In this paper, we study a well-known and influential demosaicking algorithm based upon alternating projections (AP), proposed by Gunturk, Altunbasak and Mersereau in 2002. Since its publication, the AP algorithm has been widely cited and compared against in a series of more recent papers in the demosaicking literature. Despite good performances, a limitation of the AP algorithm is its high computational complexity. We provide three main contributions in this paper. First, we present a rigorous analysis of the convergence property of the AP demosaicking algorithm, showing that it is a contraction mapping, with a unique fixed point. Second, we show that this fixed point is in fact the solution to a constrained quadratic minimization problem, thus, establishing the optimality of the AP algorithm. Finally, using the tool of polyphase representation, we show how to obtain the results of the AP algorithm in a single step, implemented as linear filtering in the polyphase domain. Replacing the original iterative procedure by the proposed one-step solution leads to substantial computational savings, by about an order of magnitude in our experiments.

Index Terms—Alternating projections, color filter array, contraction mapping, demosaicking, demosaicking, fixed point, multirate signal processing, polyphase representation, projection onto convex sets (POCS).

I. INTRODUCTION

MOST digital cameras use a single monochromatic image sensor to capture the incoming light intensities. To acquire color information, a color filter array (CFA) is placed

Manuscript received June 26, 2009; revised February 06, 2010. First published March 15, 2010; current version published July 16, 2010. This work was supported in part by the Swiss National Science Foundation under grant 200021–121935, and in part by the NCCR MICS internship program for female undergraduate students. This paper follows the concepts of reproducible research. All the results and examples presented in the paper are reproducible using the code and images available online at <http://rr.epfl.ch/demosaicking>. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Maya R. Gupta.

Y. M. Lu was with the Audiovisual Communications Laboratory, School of Computer and Communication Sciences, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland. He is now with the School of Engineering and Applied Sciences, Harvard University, Cambridge MA 02138, USA (e-mail: yuelu@seas.harvard.edu).

M. Karzand was with the School of Engineering, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland. She is now with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02142 USA (e-mail: mkarzand@mit.edu).

M. Vetterli is with the Audiovisual Communications Laboratory, School of Computer and Communication Sciences, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, and with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA (e-mail: martin.vetterli@epfl.ch).

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.2010.2045710

above the image sensor, ensuring that each pixel in the array detects one of the primary color (e.g., red, green, and blue) signals. A key process in the digital imaging pipeline is, thus, to reconstruct full-resolution color signals from their CFA downsampled versions. The feasibility and quality of this interpolation process, often referred to as *demosaicking* (or demosaicing), largely depend upon the existence of strong correlations between different color channels.

Since the birth of digital cameras in the early 1990s, numerous algorithms have been proposed for image demosaicking (see [1]–[4] for excellent reviews on existing approaches). In this paper, we study one well-known algorithm proposed by Gunturk, Altunbasak and Mersereau [5], which is based upon the concept of alternating projections (AP). Published in 2002, the AP algorithm has since become one of the representative and influential methods on demosaicking, been widely cited and compared against in a series of more recent papers (e.g., [6]–[13], to name a few). Despite good performances, a limitation of the AP algorithm is its high computational complexity: The full-resolution color images are reconstructed in an iterative fashion, with each iteration involving a 2-D *nonsubsampled* subband decomposition and reconstruction of the entire image.

After a brief overview of the original AP algorithm in Section II, we present three main contributions in the current paper.

- 1) **Proof of convergence:** We provide in Section III a rigorous analysis of the convergence of the AP algorithm. In the original paper [5], the authors attribute the convergence property to projection onto convex sets. In our analysis however, we show that a more accurate explanation is based upon the contraction mapping theorem [14], with the convergence value being the unique fixed point of the mapping. Due to the popularity of the AP algorithm in the demosaicking literature, we believe that such a careful treatment is worthwhile in its own right, be it only for pedagogical reasons.
- 2) **Optimality:** We show in Section IV that the fixed point reached by the AP algorithm is the unique solution to a constrained quadratic minimization problem. This result establishes the optimality of the AP algorithm, and points out its connections to other optimization-based demosaicking schemes in the literature (e.g., [15], [16]).
- 3) **Fast algorithm:** Using the tool of polyphase signal representation [17], [18], we show in Section V how to obtain the results of the AP algorithm in a single step, implemented as linear filtering operations in the polyphase domain. Numerical experiments verify that the proposed one-step implementation achieves the same results obtained by

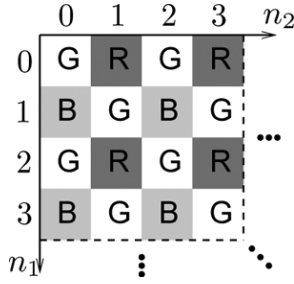


Fig. 1. Portion of the Bayer color filter array. R, G, and B represent the red, green, and blue color filters, respectively.

the original AP algorithm, but is about an order of magnitude faster, due to the elimination of the iterative procedure.

Notations: We summarize in the following the main notations used in the paper. First, $\ell^2(\mathbb{Z}^2)$ denotes the linear space of square-summable 2-D sequences. We use $x[\mathbf{n}]$, $y[\mathbf{n}]$, $r[\mathbf{n}]$, $g[\mathbf{n}]$, $b[\mathbf{n}]$, and $s[\mathbf{n}]$, with $\mathbf{n} \stackrel{\text{def}}{=} (n_1, n_2)$, to represent scalar-valued images of size $M \times N$. By filling zeros outside of their finite spatial support, all these image can be modeled as vectors in $\ell^2(\mathbb{Z}^2)$. When there is no confusion, we will drop the dependence upon the variable \mathbf{n} and use x , y , r , g , b , s for simplicity. The 2-D z -transform of x is defined as

$$X(\mathbf{z}) = \sum_{\mathbf{n} \in \mathbb{Z}^2} x[\mathbf{n}]z^{-\mathbf{n}}$$

where $\mathbf{z}^{-\mathbf{n}} \stackrel{\text{def}}{=} z_1^{-n_1}z_2^{-n_2}$. Setting $\mathbf{z} = e^{j\boldsymbol{\omega}} \stackrel{\text{def}}{=} (e^{j\omega_1}, e^{j\omega_2})$ leads to the Fourier transform, which—by abuse of notation—will be written as $X(\boldsymbol{\omega})$ instead of $X(e^{j\boldsymbol{\omega}})$. The polyphase representation plays an important role in our discussions. For any image x , we use x_{00} , x_{01} , x_{10} , and x_{11} to represent its four polyphase components, whose definitions are given in Section III-A. We denote by $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4\}$ the standard Euclidean basis vectors in \mathbb{R}^4 , and by \mathbf{I} the identity matrix in $\mathbb{R}^{d \times d}$, where the dimension d will be clear from the context. Finally, $\mathbb{1}_{\mathcal{D}}(\boldsymbol{\omega})$ denotes the indicator function of a domain \mathcal{D} , i.e., $\mathbb{1}_{\mathcal{D}}(\boldsymbol{\omega}) = 1$ if $\boldsymbol{\omega} \in \mathcal{D}$ and $\mathbb{1}_{\mathcal{D}}(\boldsymbol{\omega}) = 0$ otherwise.

II. BACKGROUND

We briefly overview the original AP demosaicking algorithm [5], which sets the ground for all subsequent discussions.

A. Demosaicking by Alternating Projections

Fig. 1 shows the sampling pattern of the widely used Bayer CFA [19]. Let s denote the raw sensor image obtained from the CFA, and let r , g , b represent the full-resolution red, green, and blue color channels, respectively. The goal of the AP demosaicking algorithm is to estimate these full-resolution color images from s .

Similar to many other demosaicking schemes in the literature, the AP algorithm starts by obtaining an estimate of the full-resolution green channel, which is then used in the subsequent estimation of the missing red and blue pixels. This two-step strategy is justified by the fact that the green channel in the Bayer CFA has twice the sampling density as that of the red or blue channels (see Fig. 1), and, hence, is relatively easier to reconstruct.

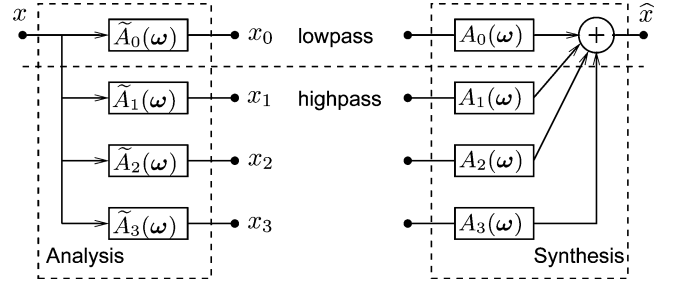


Fig. 2. 2-D nonsampled filter bank with one lowpass channel $[\tilde{A}_0(\boldsymbol{\omega})$ and $A_0(\boldsymbol{\omega})]$ and three highpass channels $[\tilde{A}_i(\boldsymbol{\omega})$ and $A_i(\boldsymbol{\omega})$ for $i = 1 \dots 3$].

In the AP algorithm, the green channel is initially estimated by an edge-directed interpolation scheme (such as [20]), followed by a subband-based update step. We omit further descriptions of this update step, whose details can be found in [5].

In this work, our focus is on how the AP algorithm iteratively interpolates the missing red and blue pixels. To avoid repetition, our discussion will be mostly concentrated on the red pixels throughout the paper. The processing steps for the blue pixels are similar, and can be easily inferred by symmetry.

In estimating the red channel, the AP algorithm alternates between two constraint sets. The first constraint exploits the correlation between the highpass (i.e., “detail”) subbands of the green and red channels. For subband decomposition, consider a 2-D nonsampled wavelet transform shown in Fig. 2. With one level of decomposition, an input image x is split into one lowpass subband x_0 and three highpass subbands x_1 , x_2 , x_3 . The set of analysis filters $\{\tilde{A}_i(\boldsymbol{\omega})\}$ and synthesis filters $\{A_i(\boldsymbol{\omega})\}$ in the filter bank satisfy the perfect reconstruction condition

$$\sum_{i=0}^3 \tilde{A}_i(\boldsymbol{\omega})A_i(\boldsymbol{\omega}) \equiv 1. \quad (1)$$

We can choose all these channel filters to be separable products of 1-D filters. For example, the lowpass analysis and synthesis filters used in [5] are

$$\begin{aligned} \tilde{a}_0[\mathbf{n}] &= \frac{1}{16} [1 \ 2 \ 1]^T [1 \ 2 \ 1], \text{ and} \\ a_0[\mathbf{n}] &= \frac{1}{64} [-1 \ 2 \ 6 \ 2 \ -1]^T [-1 \ 2 \ 6 \ 2 \ -1] \end{aligned} \quad (2)$$

respectively.

Based upon the observation [5], [21] that the highpass subbands of the green and red channels of a natural image tend to be very similar, the AP algorithm proposes to search for full-resolution red images within the following “detail constraint set”

$$\mathcal{C}_d \stackrel{\text{def}}{=} \{r[\mathbf{n}] : |(\tilde{a}_i * r)[\mathbf{n}] - (\tilde{a}_i * \hat{g})[\mathbf{n}]| \leq T, \text{ for } 1 \leq i \leq 3\} \quad (3)$$

where \tilde{a}_i is the i th highpass analysis filter in Fig. 2, \hat{g} is the previously estimated full-resolution green channel, and T is a

threshold indicating how “similar” the two signals should be. In practice, it is often sufficient to set T to zero, which works well when the red and green channels are strongly correlated.

To enforce the constraint defined in (3) (with $T = 0$), the AP algorithm employs a “detail update operator” $P_d : \ell^2(\mathbb{Z}^2) \mapsto \ell^2(\mathbb{Z}^2)$, defined in the transform domain as

$$y = P_d x \Rightarrow Y(\boldsymbol{\omega}) = A_0(\boldsymbol{\omega})\tilde{A}_0(\boldsymbol{\omega})X(\boldsymbol{\omega}) + \sum_{i=1}^3 A_i(\boldsymbol{\omega})\tilde{A}_i(\boldsymbol{\omega})\hat{G}(\boldsymbol{\omega}). \quad (4)$$

The previously shown formulation can be intuitively understood by referring to Fig. 2. For any input image $X(\boldsymbol{\omega})$, the operator P_d takes the lowpass subband of X [i.e., $\tilde{A}_0(\boldsymbol{\omega})X(\boldsymbol{\omega})$] and the highpass subbands of the previously estimated green channel \hat{G} [i.e., $\tilde{A}_i(\boldsymbol{\omega})\hat{G}(\boldsymbol{\omega})$], and then combine them through the synthesis filter bank to get the output $Y(\boldsymbol{\omega})$.

We observe that formula (4) for the detail update operator can be simplified as follows. Introduce a lowpass filter

$$L(\boldsymbol{\omega}) \stackrel{\text{def}}{=} \tilde{A}_0(\boldsymbol{\omega})A_0(\boldsymbol{\omega}). \quad (5)$$

It then follows from the perfect reconstruction property (1) of the filter bank that

$$1 - L(\boldsymbol{\omega}) = \sum_{i=1}^3 \tilde{A}_i(\boldsymbol{\omega})A_i(\boldsymbol{\omega}). \quad (6)$$

Substitute (5) and (6) into (4)

$$y = P_d x \Rightarrow Y(\boldsymbol{\omega}) = L(\boldsymbol{\omega})X(\boldsymbol{\omega}) + (1 - L(\boldsymbol{\omega}))\hat{G}(\boldsymbol{\omega}). \quad (7)$$

Compared with the original definition (4) for P_d , the proposed formula in (7) only requires the filtering of $X(\boldsymbol{\omega})$ and $\hat{G}(\boldsymbol{\omega})$ in the lowpass channel of the filter bank. Consequently, we can completely eliminate the computations previously needed in obtaining the three highpass channels of $\hat{G}(\boldsymbol{\omega})$.

The second constraint the AP algorithm employs is based upon the available sensor measurements s . Specifically, the interpolated red channel should belong to the following “observation constraint set:”

$$\mathcal{C}_o \stackrel{\text{def}}{=} \{r[\mathbf{n}] : r[\mathbf{n}] = s[\mathbf{n}], \text{ for } \mathbf{n} \in \Lambda_r\} \quad (8)$$

where $\Lambda_r \stackrel{\text{def}}{=} \{\mathbf{n} = (2k_1, 2k_2 + 1) : k_1, k_2 \in \mathbb{Z}\}$ represents the locations of the red pixels in the Bayer CFA shown in Fig. 1. To enforce the previously mentioned consistency requirement, we can define the “observation update operator” $P_o : \ell^2(\mathbb{Z}^2) \mapsto \ell^2(\mathbb{Z}^2)$ as

$$y[\mathbf{n}] = (P_o x)[\mathbf{n}] \stackrel{\text{def}}{=} \begin{cases} s[\mathbf{n}], & \text{if } \mathbf{n} \in \Lambda_r \\ x[\mathbf{n}], & \text{otherwise.} \end{cases} \quad (9)$$

In words, the operator P_o replaces the values of $x[\mathbf{n}]$ at the red pixel locations with the available sensor measurements, but leaves the rest of the pixels intact.

After defining the two update operators P_o and P_d , we can now summarize the main iterations of the AP algorithm.

Algorithm 1 Interpolate the Missing Red Pixels by Alternating Projections [5]

Input: The sensor image s from the Bayer CFA, and the estimated green channel \hat{g} .

Output: An estimated full-resolution red channel \hat{r} of $M \times N$ pixels.

Use bilinear interpolation to obtain an initial estimate $r^{(0)}$ of the red channel.

Initialize the iteration number: $k \leftarrow 0$

repeat

Enforce the detail constraint by having $r^{(k+0.5)} = P_d r^{(k)}$.

Enforce the observation constraint by having $r^{(k+1)} = P_o r^{(k+0.5)}$.

$k \leftarrow k + 1$

until The mean squared error $\text{MSE} \stackrel{\text{def}}{=} \frac{\|r^{(k)} - r^{(k-1)}\|^2}{(MN)}$ is smaller than a given threshold δ .

return $\hat{r} = r^{(k)}$

B. Convergence Property

Starting from an initial estimate $r^{(0)}$, the AP algorithm described previously generates a sequence of updated estimates $\{r^{(1)}, r^{(2)}, r^{(3)}, \dots\}$, where

$$r^{(k+1)} = P_o P_d r^{(k)}, \quad \text{for } k = 0, 1, 2, \dots$$

Numerical experiments indicate that this iterative procedure converges within a small tolerance ($\text{MSE} < \delta = 0.2$) after about 5 to 7 iterations. In [5], the authors attribute this desirable convergence property to the classical *alternating projection theorem*. To have a rigorous convergence analysis, we first recall the following facts about projections onto closed convex sets (POCS).

Definition 1 (Projection): Let \mathcal{C} be a closed convex set in a Hilbert space \mathcal{H} . For any $x \in \mathcal{H}$, there exists a *unique* element $y_x \in \mathcal{C}$ such that

$$\|x - y_x\| \leq \|x - z\|, \quad \text{for all } z \in \mathcal{C}.$$

We call the mapping $P_{\mathcal{C}} : \mathcal{H} \mapsto \mathcal{H}$, $P_{\mathcal{C}} x \stackrel{\text{def}}{=} y_x$ the *projection operator* onto \mathcal{C} .

Theorem 1 (Alternating Projections [22]): Let \mathcal{C}_1 and \mathcal{C}_2 be two closed convex sets in a Hilbert space \mathcal{H} , and $P_{\mathcal{C}_1}$ and $P_{\mathcal{C}_2}$ the corresponding projection operators. Suppose that $\mathcal{C}_1 \cap \mathcal{C}_2 \neq \emptyset$. For any $x^{(0)} \in \mathcal{H}$, the sequence $\{x^{(k+1)} : x^{(k+1)} = P_{\mathcal{C}_1} P_{\mathcal{C}_2} x^{(k)}\}_{k=0,1,2,\dots}$ converges to a point $x^* \in \mathcal{C}_1 \cap \mathcal{C}_2$.

To invoke the previously mentioned theorem in the context of the AP algorithm, one can easily verify that the two constraint sets \mathcal{C}_o and \mathcal{C}_d defined in Section II-A are closed and convex. Meanwhile, the observation operator P_o in (9) is indeed the projection (i.e., best approximation) onto \mathcal{C}_o . However, this is in

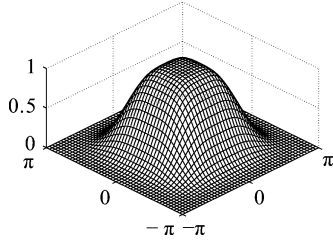


Fig. 3. Magnitude frequency response of the lowpass filter $L(\omega)$ used in [5]. It has a smooth transition band and, hence, does not satisfy (10).

general not the case for the detail update operator P_d defined in (7).

Proposition 1: P_d is a projection operator if and only if

$$L(\omega) = \mathbb{1}_{\mathcal{D}}(\omega) \quad (10)$$

where $L(\omega)$ is the lowpass filter defined in (5), and $\mathbb{1}_{\mathcal{D}}(\omega)$ is the indicator function of the frequency domain support \mathcal{D} of $L(\omega)$.

Proof: To show the necessity of (10), we suppose that P_d is a projection. Recall that a well-known property of projection operators is *idempotence* [23], i.e., $P_d P_d = P_d$. From the definition of P_d in (7), we must have, for any input image $X(\omega)$

$$\begin{aligned} L(\omega) \left(L(\omega)X(\omega) + (1 - L(\omega))\widehat{G}(\omega) \right) \\ + (1 - L(\omega))\widehat{G}(\omega) = L(\omega)X(\omega) + ((1 - L(\omega))\widehat{G}(\omega)). \end{aligned}$$

Rearranging the equality leads to

$$L(\omega)(1 - L(\omega)) \left(\widehat{G}(\omega) - X(\omega) \right) = 0. \quad (11)$$

Since (11) holds for all possible input $X(\omega)$, we must have

$$L(\omega)(1 - L(\omega)) = 0$$

which implies that $L(\omega)$ has a binary-valued ideal frequency response, i.e., $L(\omega) = 1$ for ω in the passband \mathcal{D} and $L(\omega) = 0$ otherwise. The sufficiency of (10) is shown in Appendix A. ■

Proposition 1 states that, for the detail update operator P_d to be a projection, the lowpass filter $L(\omega)$ must be ideal in the frequency domain, and correspondingly, must have an infinitely-supported sinc-like impulse response in space. This requirement is clearly not satisfied in practical implementations of the AP algorithm, where filters with finite impulse responses (FIR) are used. For example, we show in Fig. 3 the magnitude frequency response

$$|L(\omega)| = \left| \widetilde{A}_0(\omega)A_0(\omega) \right|$$

resulting from the two lowpass filters $\widetilde{A}_0(\omega)$ and $A_0(\omega)$ used in [5] (see (2) for their spatial-domain specifications). Evidently, $|L(\omega)|$ is nonideal in frequency and consequently, P_d is not a projection operator. Nevertheless, we know from numerical experiments that the AP algorithm based on this filter still converges after several iterations.

The previous discussions indicate that only when using ideal filters can we attribute the convergence of the AP algorithm to POCS (i.e., Theorem 1). When using nonideal FIR filters (which

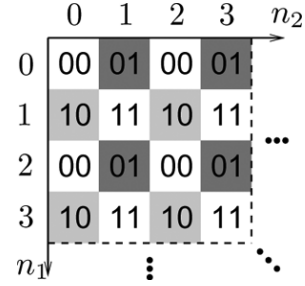


Fig. 4. Sampling locations of the four polyphase components of a 2-D signal.

is always the case in practice), we need to seek a different and more rigorous explanation for the convergence property of the AP algorithm. This is the focus of the following section.

III. CONVERGENCE OF THE AP ALGORITHM

In this section, we show that, for suitable choices of the lowpass filter $L(\omega)$, the AP algorithm is in fact a contraction mapping, with a unique fixed point. Important to our theoretical derivation is the *polyphase representation* [17], [18] of signals, which is a powerful tool widely used in multirate signal processing. To help readers who are not familiar with this concept, we start our discussion with a brief overview of its definition and main properties.

A. Polyphase Representation

In this paper, we will use the following form of 2-D polyphase decomposition.

Definition 2: A 2-D image (or filter) $x[\mathbf{n}]$ can be split into four nonoverlapping *polyphase components* $x_{00}[\mathbf{n}]$, $x_{01}[\mathbf{n}]$, $x_{10}[\mathbf{n}]$, and $x_{11}[\mathbf{n}]$, defined as

$$x_{i,j}[n_1, n_2] \stackrel{\text{def}}{=} x[2n_1 + i, 2n_2 + j], \quad \text{for } i, j \in \{0, 1\}. \quad (12)$$

As illustrated in Fig. 4, the polyphase components specified in (12) are simply downsampled versions of the original signal x . The sampling locations of all four polyphase components form a complete partition. Note that the mapping between the signal x and its polyphase components is one-to-one. To reconstruct the original signal from its polyphase components, we can easily verify that, in the z -domain

$$\begin{aligned} X(\mathbf{z}) = X_{00}(\mathbf{z}^2) + z_2^{-1}X_{01}(\mathbf{z}^2) \\ + z_1^{-1}X_{10}(\mathbf{z}^2) + z_1^{-1}z_2^{-1}X_{11}(\mathbf{z}^2). \end{aligned} \quad (13)$$

The polyphase representation becomes especially handy when we want to describe the periodic sampling structures of color filter arrays. For example, the sampling patterns of the Bayer CFA (see Fig. 1) can be succinctly described in the polyphase domain as follows:

$$s_{00} = g_{00}, s_{01} = r_{01}, s_{10} = b_{10}, \text{ and } s_{11} = g_{11}$$

where s represents the raw sensor image.

Definition 3: In the rest of the paper, we denote by

$$\xi_X(\omega) \stackrel{\text{def}}{=} [X_{00}(\omega), X_{01}(\omega), X_{10}(\omega), X_{11}(\omega)]^T \quad (14)$$

the (Fourier domain) *polyphase vector* of an image $X(\boldsymbol{\omega})$, and by (15), shown at the bottom of the page, the *polyphase matrix* of a filter $L(\boldsymbol{\omega})$.

Lemma 1: Consider two images $X(\boldsymbol{\omega})$, $Y(\boldsymbol{\omega})$ such that

$$Y(\boldsymbol{\omega}) = L(\boldsymbol{\omega})X(\boldsymbol{\omega}) \quad (16)$$

for some filter $L(\boldsymbol{\omega})$. Then

$$\boldsymbol{\xi}_Y(\boldsymbol{\omega}) = \Phi_L(\boldsymbol{\omega})\boldsymbol{\xi}_X(\boldsymbol{\omega}). \quad (17)$$

Proof: See Appendix B. ■

At this point, it may seem a bit counterproductive to work in the polyphase domain, since the simple filtering operation in (16) becomes a more complicated matrix-vector multiplication in (17). As we shall see in the following, however, the primary advantage in using the polyphase notation is that it can convert the *shift-variant*¹ operator $P_o P_d$ used in the AP algorithm to a multiple-input, multiple-output (MIMO) system of *shift-invariant* (filtering) operations. The additional complication in dealing with matrix-vector multiplications will be more than compensated for by the convenience of working with shift-invariant operators.

B. Simplifying the AP Algorithm in the Polyphase Domain

Equipped with the tool of polyphase representation, we can now express the iterative procedure

$$r^{(k+1)} = P_o r^{(k+0.5)} = P_o P_d r^{(k)}, \quad \text{for } k = 0, 1, 2, \dots$$

defined in Algorithm 1 in a simplified form. To that end, we first introduce the following change of variable:

$$c \stackrel{\text{def}}{=} r - \hat{g} \quad (18)$$

where \hat{g} is the estimated green channel, and r can be either $r^{(k)}$ or $r^{(k+0.5)}$. We refer to c as the ‘‘chrominance’’ signal, since it is the difference between the red and green channels. Note that studying c will be equivalent to studying r , because \hat{g} is estimated before the iteration process and remains constant.

As shown in the following lemma, the main advantage in working with the chrominance signal in (18) is the simplification of the detail update operator P_d from the form of (7) to a *single* filtering operation.

Lemma 2: After the detail update step at the k th iteration, we have

$$C^{(k+0.5)}(\boldsymbol{\omega}) = L(\boldsymbol{\omega})C^{(k)}(\boldsymbol{\omega})$$

¹We can easily verify that the observation update operator P_o as defined in (9) is shift-variant.

and

$$\boldsymbol{\xi}_{C^{(k+0.5)}}(\boldsymbol{\omega}) = \Phi_L(\boldsymbol{\omega})\boldsymbol{\xi}_{C^{(k)}}(\boldsymbol{\omega}) \quad (19)$$

where $\boldsymbol{\xi}_{C^{(k)}}(\boldsymbol{\omega})$ and $\boldsymbol{\xi}_{C^{(k+0.5)}}(\boldsymbol{\omega})$ are the polyphase vectors of $C^{(k)}(\boldsymbol{\omega})$ and $C^{(k+0.5)}(\boldsymbol{\omega})$, respectively; and $\Phi_L(\boldsymbol{\omega})$ is the polyphase matrix of the filter $L(\boldsymbol{\omega})$.

Proof: By definition, $c^{(k+0.5)} = r^{(k+0.5)} - \hat{g} = (P_d r^{(k)}) - \hat{g}$. Using the formula (7) for P_d yields

$$\begin{aligned} C^{(k+0.5)}(\boldsymbol{\omega}) &= L(\boldsymbol{\omega})R^{(k)}(\boldsymbol{\omega}) + (1 - L(\boldsymbol{\omega}))\hat{G}(\boldsymbol{\omega}) - \hat{G}(\boldsymbol{\omega}) \\ &= L(\boldsymbol{\omega})\left(R^{(k)}(\boldsymbol{\omega}) - \hat{G}(\boldsymbol{\omega})\right) \\ &= L(\boldsymbol{\omega})C^{(k)}(\boldsymbol{\omega}). \end{aligned}$$

The equality in (19) can be obtained by applying Lemma 1. ■

Next, we focus on the observation update operator P_o . From the definition of P_o in (9) and that of the polyphase components in (12), it is straightforward to verify the following result, describing P_o in the polyphase domain.

Lemma 3: After the observation update step at the k th iteration, we have

$$c_{ij}^{(k+1)} = \begin{cases} s_{01} - \hat{g}_{01}, & \text{if } i = 0, j = 1 \\ c_{ij}^{(k+0.5)}, & \text{otherwise.} \end{cases} \quad (20)$$

The key observation from (20) is that one of the polyphase components $c_{01}^{(k)}$ remains the same (equal to $s_{01} - \hat{g}_{01}$) during the entire iteration process. Consequently, we only need to study the evolution and convergence of the three remaining polyphase components.

Definition 4: We denote by

$$\boldsymbol{\theta}_X(\boldsymbol{\omega}) \stackrel{\text{def}}{=} [X_{00}(\boldsymbol{\omega}), X_{10}(\boldsymbol{\omega}), X_{11}(\boldsymbol{\omega})]^T \quad (21)$$

the transform domain *partial polyphase vector* of an image $X(\boldsymbol{\omega})$.

Let $\{\mathbf{e}_i\}_{i=1}^4$ be the standard Euclidean basis vectors in \mathbb{R}^4 , and let

$$\mathbf{M} \stackrel{\text{def}}{=} [\mathbf{e}_1, \mathbf{e}_3, \mathbf{e}_4]. \quad (22)$$

We can easily verify the following identity relating the *full* and *partial* polyphase vectors:

$$\boldsymbol{\theta}_X(\boldsymbol{\omega}) = \mathbf{M}^T \boldsymbol{\xi}_X(\boldsymbol{\omega}). \quad (23)$$

$$\Phi_L(\boldsymbol{\omega}) \stackrel{\text{def}}{=} \begin{bmatrix} L_{00}(\boldsymbol{\omega}) & L_{01}(\boldsymbol{\omega}) e^{-j\omega_2} & L_{10}(\boldsymbol{\omega}) e^{-j\omega_1} & L_{11}(\boldsymbol{\omega}) e^{-j(\omega_1+\omega_2)} \\ L_{01}(\boldsymbol{\omega}) & L_{00}(\boldsymbol{\omega}) & L_{11}(\boldsymbol{\omega}) e^{-j\omega_1} & L_{10}(\boldsymbol{\omega}) e^{-j\omega_1} \\ L_{10}(\boldsymbol{\omega}) & L_{11}(\boldsymbol{\omega}) e^{-j\omega_2} & L_{00}(\boldsymbol{\omega}) & L_{01}(\boldsymbol{\omega}) e^{-j\omega_2} \\ L_{11}(\boldsymbol{\omega}) & L_{10}(\boldsymbol{\omega}) & L_{01}(\boldsymbol{\omega}) & L_{00}(\boldsymbol{\omega}) \end{bmatrix} \quad (15)$$

Meanwhile, for the chrominance signal $C^{(k)}(\boldsymbol{\omega})$, it follows from Lemma 3 that:

$$\boldsymbol{\xi}_{C^{(k)}}(\boldsymbol{\omega}) = \mathbf{M}\boldsymbol{\theta}_{C^{(k)}}(\boldsymbol{\omega}) + \mathbf{e}_2 \left(S_{01}(\boldsymbol{\omega}) - \widehat{G}_{01}(\boldsymbol{\omega}) \right). \quad (24)$$

Proposition 2: In the AP algorithm, the $(k+1)$ th estimate $\boldsymbol{\theta}_{C^{(k+1)}}(\boldsymbol{\omega})$ is obtained from the previous one $\boldsymbol{\theta}_{C^{(k)}}(\boldsymbol{\omega})$ through a fixed linear mapping T , defined as

$$\begin{aligned} \boldsymbol{\theta}_{C^{(k+1)}}(\boldsymbol{\omega}) &= (T\boldsymbol{\theta}_{C^{(k)}})(\boldsymbol{\omega}) \\ &\stackrel{\text{def}}{=} \mathbf{T}_L(\boldsymbol{\omega})\boldsymbol{\theta}_{C^{(k)}}(\boldsymbol{\omega}) \\ &\quad + \mathbf{M}^T\boldsymbol{\Phi}_L(\boldsymbol{\omega})\mathbf{e}_2 \left(S_{01}(\boldsymbol{\omega}) - \widehat{G}_{01}(\boldsymbol{\omega}) \right) \end{aligned} \quad (25)$$

where $\mathbf{T}_L(\boldsymbol{\omega}) \stackrel{\text{def}}{=} \mathbf{M}^T\boldsymbol{\Phi}_L(\boldsymbol{\omega})\mathbf{M}$, and $\boldsymbol{\Phi}_L(\boldsymbol{\omega})$ is the polyphase matrix of the lowpass filter $L(\boldsymbol{\omega})$.

Remark: By construction, both $\mathbf{T}_L(\boldsymbol{\omega})$ and $\mathbf{M}^T\boldsymbol{\Phi}_L(\boldsymbol{\omega})\mathbf{e}_2$ in (25) are submatrices of $\boldsymbol{\Phi}_L(\boldsymbol{\omega})$. The former is obtained by retaining rows 1, 3, 4 and columns 1, 3, 4; the latter by rows 1, 3, 4 and column 2.

Proof: Starting from Lemma 3, we have

$$\begin{aligned} \boldsymbol{\theta}_{C^{(k+1)}}(\boldsymbol{\omega}) &= \boldsymbol{\theta}_{C^{(k+0.5)}}(\boldsymbol{\omega}) \\ &= \mathbf{M}^T\boldsymbol{\xi}_{C^{(k+0.5)}}(\boldsymbol{\omega}) \\ &= \mathbf{M}^T\boldsymbol{\Phi}_L(\boldsymbol{\omega})\boldsymbol{\xi}_{C^{(k)}}(\boldsymbol{\omega}). \end{aligned} \quad (26)$$

The second equality mentioned previously is due to (23) (replacing X by $C^{(k+0.5)}$), and the last equality is due to Lemma 2. Substituting (24) into (26) yields

$$\begin{aligned} \boldsymbol{\theta}_{C^{(k+1)}}(\boldsymbol{\omega}) &= \mathbf{M}^T\boldsymbol{\Phi}_L(\boldsymbol{\omega}) \left(\mathbf{M}\boldsymbol{\theta}_{C^{(k)}}(\boldsymbol{\omega}) + \mathbf{e}_2 \left(S_{01}(\boldsymbol{\omega}) - \widehat{G}_{01}(\boldsymbol{\omega}) \right) \right) \end{aligned}$$

and, hence, (25).

C. Convergence by Contraction Mapping

Having expressed each iteration of the AP algorithm as a fixed linear mapping T in Proposition 2, we are now ready to present one of the key results of this work: the convergence of the AP algorithm based upon contraction mapping. We first recall the following facts relevant to our discussions.

Definition 5: A mapping T from a Hilbert space to itself is called *Lipschitz continuous*, if there is some real number $\alpha > 0$ such that

$$\|Tx_1 - Tx_2\| \leq \alpha\|x_1 - x_2\|, \quad \text{for all } x_1, x_2 \in \mathcal{H}.$$

The smallest such value of α , denoted by α_T , is called the *Lipschitz constant* of T . Furthermore, if $\alpha_T < 1$, then T is a *contraction mapping*.

Theorem 2 (Contraction Mapping [14, pp. 272–273]): Let T be a contraction mapping with a Lipschitz constant $0 < \alpha_T < 1$.

- 1) The mapping T admits one and only one fixed point \hat{x} , i.e., $T\hat{x} = \hat{x}$.
- 2) For arbitrary $x^{(0)} \in \mathcal{H}$, the sequence $x^{(k+1)} = Tx^{(k)}$ always converges to this fixed point \hat{x} .

- 3) The speed of convergence is bounded by the following inequality:

$$\|x^{(k)} - \hat{x}\| \leq (\alpha_T)^k \|x^{(0)} - \hat{x}\|. \quad (27)$$

Now if we can show that the mapping T as defined in (25) is indeed a contraction, then the convergence of the AP algorithm will be automatically guaranteed by the contraction mapping theorem stated previously. To that end, the first step is to obtain the Lipschitz constant α_T . Since T represents a MIMO filtering operation, we show in Appendix C that α_T can be easily calculated in the Fourier domain as

$$\alpha_T = \max_{\boldsymbol{\omega}} \sigma_1(\mathbf{T}_L(\boldsymbol{\omega})) \quad (28)$$

where $\mathbf{T}_L(\boldsymbol{\omega})$ is a 3×3 submatrix of the polyphase matrix $\boldsymbol{\Phi}_L(\boldsymbol{\omega})$ as defined in Proposition 2, and $\sigma_1(\cdot)$ denotes the largest singular value of any given matrix.

Theorem 3 (Convergence of the AP Algorithm): The iterative procedure described in Algorithm 1 converges for arbitrary choices of the initial estimate $r^{(0)}$ if

$$\max_{\boldsymbol{\omega}} \sigma_1(\mathbf{T}_L(\boldsymbol{\omega})) < 1. \quad (29)$$

Proof: The result follows immediately from (28) (see Appendix C) and Theorem 2. \blacksquare

Remark: When $\max_{\boldsymbol{\omega}} \sigma_1(\mathbf{T}_L(\boldsymbol{\omega})) \geq 1$, the iterative AP algorithm can diverge or run into repetitive cycles. A rigorous discussion of these cases is beyond the scope of the current paper.

In practice, given an FIR filter $L(\boldsymbol{\omega})$ used in the AP algorithm, the Lipschitz constant in (28) can be computed as follows. First, we construct the polyphase matrix $\boldsymbol{\Phi}_L(\boldsymbol{\omega})$ as in (15), and then its 3×3 submatrix $\mathbf{T}_L(\boldsymbol{\omega})$ as in Proposition 2. Since the original filter $L(\boldsymbol{\omega})$ is FIR, all of the polyphase components in $\mathbf{T}_L(\boldsymbol{\omega})$ must necessarily be FIR as well and, hence, be spatially supported within a $K \times K$ window, for some $K > 0$.

Now choose a positive integer $J \geq K$. For any of the polyphase filters, we zero-pad the filter to size $J \times J$ and then apply a 2-D discrete Fourier transform (via FFT). Doing so yields the Fourier transform of that polyphase filter on a discrete grid $\{\boldsymbol{\omega} = 2\pi\mathbf{k}/J : \text{for } \mathbf{k} \in [0, 1, \dots, J-1]^2\}$. From these values we can obtain the following finite estimate

$$\alpha_T(J) = \max_{\mathbf{k} \in [0, 1, \dots, J-1]^2} \sigma_1 \left(\mathbf{T}_L \left(\frac{2\pi\mathbf{k}}{J} \right) \right)$$

which approaches the true quantity in (28) as J goes to infinity.

1) *Example 1:* Consider the two lowpass filters in (2). The corresponding product filter $L(\boldsymbol{\omega})$ is 7×7 and, hence, the polyphase filters in (15) can all be bounded within a window of size 4×4 . Using the previously mentioned estimation procedure and choosing $J = 2048$ leads to

$$\alpha_T \approx \alpha_T(2048) = 0.75 < 1.$$

It then follows from Theorem 3 that the AP algorithm based upon these filters is a contraction mapping and thus always converges. Furthermore, we know from Theorem 2 that the con-

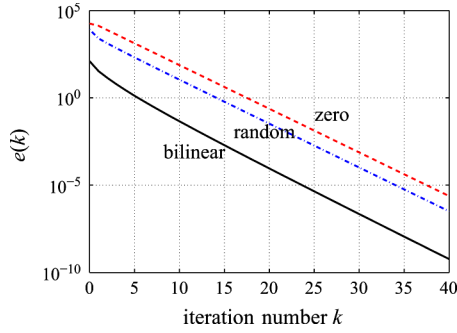


Fig. 5. Convergence of the AP algorithm. Different choices of the initial estimates (solid line: bilinear interpolation, dash-dotted line: random, dashed line: all zero) all converge to the same value, and at the same speed (slope).

vergence value is the unique fixed point of T , which does not depend upon the initial value. To verify this property, we apply the AP algorithm to the test image “light house” from the standard Kodak Photo CD image set, with three different initial estimates for $r^{(0)}$: bilinear interpolation as in [5], all zeros, and i.i.d. random numbers uniformly distributed from 0 to 255. For each initial estimate, we carry out the iteration process and calculate

$$e(k) \stackrel{\text{def}}{=} \frac{\|r^{(k)} - \hat{r}_{\text{BL}}\|^2}{(MN)}$$

where $r^{(k)}$ is the estimated red channel at the k th iteration, \hat{r}_{BL} is the convergence value (after 60 iterations) obtained by choosing bilinear interpolation as the starting point, and MN is the total number of pixels. We see from Fig. 5 that the three choices of the initial estimates all eventually converge to the same result \hat{r}_{BL} . Meanwhile, the three curves in Fig. 5 have the same rate of convergence (i.e., slope), which is determined by the Lipschitz constant α_T as specified in (27).

2) *Example 2:* There exist FIR filters for which the condition (29) on singular values does not hold. For example, let $l[\mathbf{n}]$ be a 3×3 filter whose coefficients are

$$\frac{1}{24} \begin{bmatrix} -1 & 2 & -1 \\ 2 & 20 & 2 \\ -1 & 2 & -1 \end{bmatrix}.$$

In the Fourier domain, the filter is lowpass and has a maximum response at $\boldsymbol{\omega} = \mathbf{0}$. Meanwhile, its coefficients are normalized to sum to unity. We can construct the corresponding polyphase matrix $\Phi_L(\boldsymbol{\omega})$ as in (15) and then the submatrix $T_L(\boldsymbol{\omega})$ according to Proposition 2. In particular, at $\boldsymbol{\omega} = \mathbf{0}$, we have

$$T_L(\mathbf{0}) = \frac{1}{6} \begin{bmatrix} 5 & 1 & -1 \\ 1 & 5 & 1 \\ -1 & 1 & 5 \end{bmatrix}.$$

The largest singular value of this matrix is equal to one. Therefore, for this particular filter, the contraction mapping condition in (29) is not satisfied.

D. Blue Channel

So far, our discussions on the AP algorithm have been exclusively focused on the red channel. The analysis for the blue

channel is very similar. In fact, following essentially the same derivations in Sections III-B and III-C, we can show that the iterative procedure for estimating the blue pixels converges if

$$\max_{\boldsymbol{\omega}} \sigma_1(\tilde{T}_L(\boldsymbol{\omega})) < 1 \quad (30)$$

where

$$\tilde{T}_L(\boldsymbol{\omega}) \stackrel{\text{def}}{=} \tilde{M}^T \Phi_L(\boldsymbol{\omega}) \tilde{M} \quad \text{and} \quad \tilde{M} \stackrel{\text{def}}{=} [e_1, e_2, e_4].$$

Similar to $T_L(\boldsymbol{\omega})$ in Theorem 3, $\tilde{T}_L(\boldsymbol{\omega})$ defined previously is also a submatrix of $\Phi_L(\boldsymbol{\omega})$, obtained by retaining rows 1, 2, 4, and columns 1, 2, 4.

In principle, we will need to check two conditions [i.e., (29) and (30)] to make sure that the AP algorithm converges for both the red and blue channels. In practice, however, the lowpass filter $L(\boldsymbol{\omega})$ used in the algorithm often has the following symmetric property:²

$$L(\omega_1, \omega_2) = L(\omega_2, \omega_1).$$

For such filters, we show in Appendix D that

$$\max_{\boldsymbol{\omega}} \sigma_1(T_L(\boldsymbol{\omega})) = \max_{\boldsymbol{\omega}} \sigma_1(\tilde{T}_L(\boldsymbol{\omega})).$$

Thus, the conditions (29) and (30) become equivalent, and we only need to check one of them.

IV. OPTIMALITY OF THE AP ALGORITHM

The previous section shows that, for suitably chosen filters, the iterative AP algorithm is a contraction mapping, whose convergence value is the unique fixed point of the mapping T defined in Proposition 2. In what follows, we show that this fixed point can also be formulated as the solution to a constrained quadratic minimization problem, thus establishing the optimality of the AP algorithm.

A. Optimization Formulation for Estimating the Red Channel

Consider the following constrained optimization problem:

$$\arg \min_r \|(h * (r - \hat{g}))[\mathbf{n}]\|^2 \quad \text{subject to } r_{01} = s_{01} \quad (31)$$

where $h[\mathbf{n}]$ is a given highpass filter.

As stated in Section II-A, the highpass subbands of the green and red channels of a natural image tend to be very similar. Intuitively, the goal of (31) is to find an optimal red image whose highpass subband is “closest” to that of the green image, under the constraint given by the CFA sensor observations. Similar, but more sophisticated quadratic formulations have been previously proposed in the literature, leading to several optimization-based demosaicking schemes (e.g., [15], [16]). For example, by changing the equality constraint in (31) to a quadratic penalty term, we can generalize (31) to a regularized minimization problem as in [15], which explicitly takes into account noise in the sensor measurements. In what follows, we will focus

²Such symmetry is satisfied by all separable 2-D filters in the form $L(\boldsymbol{\omega}) = L_1(\omega_1)L_1(\omega_2)$, including those used in [5].

on the original formulation in (31) for simplicity of exposition. However, the techniques used in our discussion (especially those involving the polyphase representation) can be readily applied to the more generalized formulation for the noisy situation, leading to similar convergence analysis as well as fast implementations.

Using the change of variable $c = r - \hat{g}$ introduced in (18), we first rewrite (31) as the following equivalent problem:

$$\arg \min_c \|(h * c)[\mathbf{n}]\|^2 \quad \text{subject to } c_{01} = s_{01} - \hat{g}_{01}. \quad (32)$$

Note that the equality constraint in (32) fixes one of the polyphase components of c , and thus only three polyphase components (c_{00} , c_{10} , and c_{11}) are free variables. This prompts us to work with the partial polyphase vector $\boldsymbol{\theta}_C(\boldsymbol{\omega})$ as defined in (21). Doing so converts (32) into a much simpler *unconstrained* optimization problem, as shown in the following proposition.

Proposition 3: An image c is an optimal solution to (32) if and only if $c_{01} = s_{01} - \hat{g}_{01}$ and

$$\boldsymbol{\theta}_C(\boldsymbol{\omega}) = \arg \min_{\boldsymbol{\theta}} \left\| \Phi_H(\boldsymbol{\omega}) \left(\mathbf{M}\boldsymbol{\theta} + \mathbf{e}_2 \left(S_{01}(\boldsymbol{\omega}) - \hat{G}_{01}(\boldsymbol{\omega}) \right) \right) \right\|^2 \quad (33)$$

where $\Phi_H(\boldsymbol{\omega})$ is the polyphase matrix of the highpass filter h .

Proof: Let $y[\mathbf{n}] \stackrel{\text{def}}{=} (h * c)[\mathbf{n}]$, and denote by $\boldsymbol{\xi}_Y(\boldsymbol{\omega})$ and $\boldsymbol{\xi}_C(\boldsymbol{\omega})$ the full polyphase vectors of y and c , respectively. It follows from Lemma 1 that:

$$\begin{aligned} \boldsymbol{\xi}_Y(\boldsymbol{\omega}) &= \Phi_H(\boldsymbol{\omega})\boldsymbol{\xi}_C(\boldsymbol{\omega}) \\ &= \Phi_H(\boldsymbol{\omega}) \left(\mathbf{M}\boldsymbol{\theta}_C(\boldsymbol{\omega}) + \mathbf{e}_2 \left(S_{01}(\boldsymbol{\omega}) - \hat{G}_{01}(\boldsymbol{\omega}) \right) \right) \end{aligned}$$

where the second equality is due to (24). From Parseval's theorem

$$\begin{aligned} \|(h * c)[\mathbf{n}]\|^2 &= \|y_{00}[\mathbf{n}]\|^2 + \|y_{01}[\mathbf{n}]\|^2 + \|y_{10}[\mathbf{n}]\|^2 + \|y_{11}[\mathbf{n}]\|^2 \\ &= \frac{1}{(2\pi)^2} \int_{[-\pi, \pi]^2} \|\boldsymbol{\xi}_Y(\boldsymbol{\omega})\|^2 d\boldsymbol{\omega} \quad (34) \\ &= \frac{1}{(2\pi)^2} \int_{[-\pi, \pi]^2} \left\| \Phi_H(\boldsymbol{\omega}) \left(\mathbf{M}\boldsymbol{\theta}_C(\boldsymbol{\omega}) \right. \right. \\ &\quad \left. \left. + \mathbf{e}_2 \left(S_{01}(\boldsymbol{\omega}) - \hat{G}_{01}(\boldsymbol{\omega}) \right) \right) \right\|^2 d\boldsymbol{\omega}. \quad (35) \end{aligned}$$

Accordingly, solving (32) is equivalent to minimizing the integral in (35) with respect to $\boldsymbol{\theta}_C(\boldsymbol{\omega})$. Finally, since the integrand in (35) is non-negative everywhere, we can equivalently minimize just the integrand (i.e., without taking the integration), and do so for each $\boldsymbol{\omega}$ independently. Hence, we reach (33). ■

B. Optimality of the Fixed Point Solution

Next, we establish the connection between the previously mentioned minimization problem and the fixed point solution obtained by the AP algorithm. To that end, it is required that the lowpass filter $L(\boldsymbol{\omega})$ used in the AP iteration satisfy the following conditions:

1) *Condition 1:* We choose a lowpass filter $L(\boldsymbol{\omega})$ such that³

- 1) $\max_{\boldsymbol{\omega}} \sigma_1(\mathbf{T}_L(\boldsymbol{\omega})) < 1$.
- 2) $L(\boldsymbol{\omega})$ is real-valued (i.e., a zero phase filter);
- 3) $L(\boldsymbol{\omega}) \leq 1$ for all $\boldsymbol{\omega}$ and $L(\mathbf{0}) = 1$.

The first condition guarantees the convergence of the AP algorithm (see Theorem 3); whereas the second and third conditions ensure that we can construct a filter $h[\mathbf{n}]$ whose Fourier transform $H(\boldsymbol{\omega})$ is given by

$$H(\boldsymbol{\omega}) = \sqrt{1 - L(\boldsymbol{\omega})}. \quad (36)$$

Since $L(\boldsymbol{\omega})$ is a lowpass filter, $H(\boldsymbol{\omega})$ constructed above must be a highpass filter. In particular, $H(\mathbf{0}) = 0$.

Lemma 4: The polyphase matrices of the two filters $L(\boldsymbol{\omega})$ and $H(\boldsymbol{\omega})$ defined previously are related by

$$\mathbf{I} - \Phi_L(\boldsymbol{\omega}) = \Phi_H^*(\boldsymbol{\omega})\Phi_H(\boldsymbol{\omega}). \quad (37)$$

Proof: See Appendix E. ■

Theorem 4: Assume that the lowpass filter $L(\boldsymbol{\omega})$ satisfies Condition 1, and the highpass filter $H(\boldsymbol{\omega})$ is chosen as in (36). Then the AP algorithm converges to the unique optimal solution to (31).

Proof: The convergence of the AP algorithm to a unique fixed point of a contraction mapping is guaranteed by Theorem 3. Next, we show that this fixed point is the unique minimizer of (31). By Proposition 3, we consider the equivalent, but much simpler unconstrained minimization problem in (33). Denote by $O(\boldsymbol{\theta})$ the cost function in (33), i.e.,

$$O(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \left\| \Phi_H \left(\mathbf{M}\boldsymbol{\theta} + \mathbf{e}_2 \left(S_{01} - \hat{G}_{01} \right) \right) \right\|^2.$$

For simplicity of notation, we dropped the dependence on the variable $\boldsymbol{\omega}$ in the previous definition. By expanding the squared norm

$$\begin{aligned} O(\boldsymbol{\theta}) &= \left(\boldsymbol{\theta}^* \mathbf{M}^T \Phi_H^* + \mathbf{e}_2^T \Phi_H^* \left(S_{01}^* - \hat{G}_{01}^* \right) \right) \\ &\quad \times \left(\Phi_H \mathbf{M}\boldsymbol{\theta} + \Phi_H \mathbf{e}_2 \left(S_{01} - \hat{G}_{01} \right) \right) \\ &= \boldsymbol{\theta}^* \mathbf{M}^T \Phi_H^* \Phi_H \mathbf{M}\boldsymbol{\theta} \\ &\quad + 2\text{Re} \left(\boldsymbol{\theta}^* \mathbf{M}^T \Phi_H^* \Phi_H \mathbf{e}_2 \left(S_{01} - \hat{G}_{01} \right) \right) \\ &\quad + \left\| \Phi_H \mathbf{e}_2 \left(S_{01} - \hat{G}_{01} \right) \right\|^2 \end{aligned}$$

where the superscript $*$ denotes transposition with complex conjugation.

Since $O(\boldsymbol{\theta})$ is a quadratic function, it always has a global minimum, achieved at a point denoted by $\boldsymbol{\theta}_0$. Differentiating $O(\boldsymbol{\theta})$ and setting its complex gradient [24] to zero yields

$$\begin{aligned} \mathbf{0} &= \nabla O(\boldsymbol{\theta}_0) \\ &= \mathbf{M}^T \Phi_H^* \Phi_H \mathbf{M}\boldsymbol{\theta}_0 + \mathbf{M}^T \Phi_H^* \Phi_H \mathbf{e}_2 \left(S_{01} - \hat{G}_{01} \right) \\ &= \mathbf{M}^T \left(\mathbf{I} - \Phi_L \right) \mathbf{M}\boldsymbol{\theta}_0 + \mathbf{M}^T \left(1 - \Phi_L \right) \mathbf{e}_2 \left(S_{01} - \hat{G}_{01} \right) \quad (38) \end{aligned}$$

$$= \boldsymbol{\theta}_0 - \mathbf{M}^T \Phi_L \mathbf{M}\boldsymbol{\theta}_0 - \mathbf{M}^T \Phi_L \mathbf{e}_2 \left(S_{01} - \hat{G}_{01} \right) \quad (39)$$

³We can verify that the lowpass filter used in [5] satisfies all these conditions.

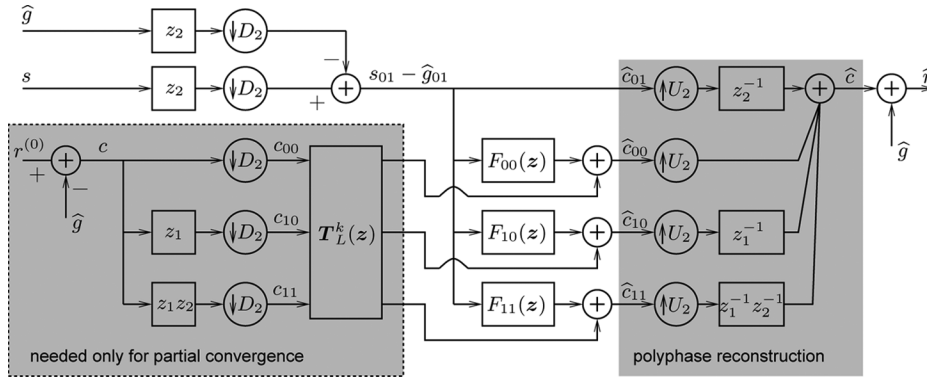


Fig. 6. Block diagram of the proposed noniterative implementation of the AP algorithm. The input signals are the estimated green channel \hat{g} , the raw CFA sensor measurement s , and in the case of partial convergence, the initial estimate of the red channel $r^{(0)}$. We use D_2 to represent downsampling by 2 along both dimensions, and U_2 upsampling by 2. To obtain the partial convergence results, the three filters $F_{00}(z)$, $F_{10}(z)$, and $F_{11}(z)$ are specified as in (41). To obtain the full convergence results, we omit the MIMO processing steps enclosed in the dashed rectangle, and change the filter specification to (46).

where in reaching (38) we use the identity given in Lemma 4. The equality in (39) implies that θ_0 is the fixed point of the mapping T defined in Proposition 2 (i.e., $\theta_0 = T\theta_0$). From the unicity of the fixed point, we can conclude that (31) admits one and only one optimal solution, which is equal to the convergence result of the AP algorithm. ■

V. EFFICIENT NONITERATIVE IMPLEMENTATION OF THE AP ALGORITHM

So far, we presented a rigorous analysis of the convergence property of the AP algorithm, and established its connection to the unique solution of an optimization formulation. We now focus on the efficient implementation of the AP algorithm.

We consider two scenarios: *partial convergence*, which corresponds to running the AP algorithm for only a small number of iterations; and *full convergence*, which represents the results obtained at convergence. Numerical experiments indicate that the partial convergence results are often slightly better, both in terms of the peak signal-to-noise ratio (PSNR) and in visual quality, than the full convergence results; whereas the advantage of the full convergence scheme, as we shall see, is in its reduced computational complexity. For each of the two cases, we propose efficient algorithms that directly obtain the desired results, without going through any iteration.

A. Directly Obtaining the Partial Convergence Results

From an initial estimate $r^{(0)}$ of the red channel (e.g., through bilinear interpolation), we obtain the chrominance signal $c^{(0)} = r^{(0)} - \hat{g}$. Decomposing $c^{(0)}$ into its polyphase components, we denote by

$$\theta_{C^{(0)}}(\omega) = [C_{00}^{(0)}(\omega), C_{10}^{(0)}(\omega), C_{11}^{(0)}(\omega)]^T$$

the partial polyphase vector of $c^{(0)}$. It follows from Proposition 2 that, at the k th iteration of the AP algorithm

$$\begin{aligned} \theta_{C^{(k)}}(\omega) &= \mathbf{T}_L^k(\omega) \theta_{C^{(0)}}(\omega) \\ &+ \left(\mathbf{T}_L^{k-1}(\omega) + \mathbf{T}_L^{k-2}(\omega) + \cdots + \mathbf{I} \right) \\ &\times \mathbf{M}^T \Phi_L(\omega) \mathbf{e}_2 \left(S_{01}(\omega) - \hat{G}_{01}(\omega) \right) \end{aligned} \quad (40)$$

where $\theta_{C^{(k)}}(\omega)$ is the partial polyphase vector of $c^{(k)}$. Assigning three filters

$$\begin{aligned} [F_{00}(\omega), F_{10}(\omega), F_{11}(\omega)] \\ = \left(\mathbf{T}_L^{k-1}(\omega) + \mathbf{T}_L^{k-2}(\omega) + \cdots + \mathbf{I} \right) \mathbf{M}^T \Phi_L(\omega) \mathbf{e}_2 \end{aligned} \quad (41)$$

we can rewrite (40) as

$$\begin{aligned} \theta_{C^{(k)}}(\omega) &= \mathbf{T}_L^k(\omega) \theta_{C^{(0)}}(\omega) \\ &+ [F_{00}(\omega), F_{10}(\omega), F_{11}(\omega)]^T \left(S_{01}(\omega) - \hat{G}_{01}(\omega) \right). \end{aligned} \quad (42)$$

This indicates that the result of the AP algorithm at the k th iteration can be directly obtained by one step of linear filtering operations in the polyphase domain.

Fig. 6 summarizes the block diagram of the proposed noniterative implementation of the AP algorithm. Starting from an initial estimate of the chrominance signal $c = r^{(0)} - \hat{g}$, we first decompose it into its polyphase components. From Lemma 3, one of the polyphase components, c_{01} , stays constant and is equal to $s_{01} - \hat{g}_{01}$. The remaining three components— c_{00} , c_{10} , and c_{11} —are updated according to (42), which involves a MIMO filtering block [i.e., a 3×3 matrix $\mathbf{T}_L^k(\omega)$] and three additional filtering operations by $F_{00}(\omega)$, $F_{10}(\omega)$, and $F_{11}(\omega)$, respectively. Finally, the desired full-resolution estimate \hat{c} can be obtained from the updated polyphase components— \hat{c}_{00} , \hat{c}_{01} , \hat{c}_{10} , and \hat{c}_{11} —through a standard polyphase reconstruction block.

B. Directly Obtaining the Full Convergence Results

If the convergence condition in Theorem 3 holds, we can also directly obtain the full convergence results of the AP algorithm. To that end, let k go to infinity in (40), and we proceed to determine the corresponding limit of $\theta_{C^{(k)}}(\omega)$.

The convergence condition (29) implies that the matrix $\mathbf{T}_L(\omega)$ is contractive for all ω . It follows that:

$$\lim_{k \rightarrow \infty} \mathbf{T}_L^k(\omega) = \mathbf{0} \quad (43)$$

and

$$\lim_{k \rightarrow \infty} \left(\mathbf{T}_L^{k-1}(\omega) + \mathbf{T}_L^{k-2}(\omega) + \cdots + \mathbf{I} \right) = (\mathbf{I} - \mathbf{T}_L(\omega))^{-1} \quad (44)$$

TABLE I
COMPARISON OF THE ITERATIVE AP ALGORITHM AND THE PROPOSED NONITERATIVE SCHEME. TO AVOID BOUNDARY ARTIFACTS, WE EXCLUDE 10 PIXELS ALONG THE BORDERS IN CALCULATING THE PSNR. THE COMPLEXITY IS GIVEN IN TERMS OF THE APPROXIMATE NUMBERS OF REQUIRED MULTIPLICATIONS AND ADDITIONS

	PSNR (dB)			CPU Time (seconds)	Multiplications	Additions
	Red	Green	Blue			
<i>Partial Convergence</i>						
AP (6 iterations)	38.54	41.80	38.58	2.459	294 <i>MN</i>	380 <i>MN</i>
Proposed	38.51	41.80	38.59	0.291	42 <i>MN</i>	75 <i>MN</i>
Ratio	—	—	—	8.5	7.0	5.1
<i>Full Convergence</i>						
AP (20 iterations)	38.44	41.80	38.41	7.011	910 <i>MN</i>	1164 <i>MN</i>
Proposed	38.42	41.80	38.39	0.173	14 <i>MN</i>	25 <i>MN</i>
Ratio	—	—	—	40.5	65.0	46.6

where (44) can be understood as a matrix generalization of the well-known Taylor series expansion $1/(1-x) = 1+x+x^2+\dots$ for $|x| < 1$. The bound on the singular value in (29) guarantees that the inverse $(\mathbf{I} - \mathbf{T}_L(\boldsymbol{\omega}))^{-1}$ always exists and is stable.

Substituting (43) and (44) into (40), we can write the full convergence result, denoted by $\boldsymbol{\theta}_{\hat{C}}(\boldsymbol{\omega})$, as

$$\boldsymbol{\theta}_{\hat{C}}(\boldsymbol{\omega}) = (\mathbf{I} - \mathbf{T}_L(\boldsymbol{\omega}))^{-1} \mathbf{M}^T \boldsymbol{\Phi}_L(\boldsymbol{\omega}) \times \mathbf{e}_2 \left(S_{01}(\boldsymbol{\omega}) - \hat{G}_{01}(\boldsymbol{\omega}) \right). \quad (45)$$

Alternatively, the previously mentioned equality can also be directly reached by using the fact that the convergence result of the AP algorithm is the unique fixed point of the mapping T defined in (25). Reassigning the three filters introduced in (41) to their respective limiting values

$$[F_{00}(\boldsymbol{\omega}), F_{10}(\boldsymbol{\omega}), F_{11}(\boldsymbol{\omega})]^T = (\mathbf{I} - \mathbf{T}_L(\boldsymbol{\omega}))^{-1} \mathbf{M}^T \boldsymbol{\Phi}_L(\boldsymbol{\omega}) \mathbf{e}_2 \quad (46)$$

the equality (45) becomes

$$\boldsymbol{\theta}_{\hat{C}}(\boldsymbol{\omega}) = [F_{00}(\boldsymbol{\omega}), F_{10}(\boldsymbol{\omega}), F_{11}(\boldsymbol{\omega})]^T \left(S_{01}(\boldsymbol{\omega}) - \hat{G}_{01}(\boldsymbol{\omega}) \right).$$

Compared with (42) for partial convergence, the formula for obtaining the full convergence result is much simpler and requires fewer computations, since it does not need to implement the MIMO block $\mathbf{T}_L^k(\boldsymbol{\omega})$ (recall that $\mathbf{T}_L^k(\boldsymbol{\omega}) \rightarrow \mathbf{0}$ as k goes to infinity). Consequently, the final result $\boldsymbol{\theta}_{\hat{C}}(\boldsymbol{\omega})$ does not depend upon the initial estimate $C^{(0)}(\boldsymbol{\omega}) = R^{(0)}(\boldsymbol{\omega}) - \hat{G}(\boldsymbol{\omega})$. This interesting fact is theoretically guaranteed by the property of contraction mapping (see Theorem 2) and has been numerically verified in Example 1.

C. Implementation and Numerical Experiments

To demonstrate the performance of the proposed scheme shown in Fig. 6, we apply both the proposed noniterative algorithm and the original iterative AP algorithm to the 24 standard Kodak test images. All the MATLAB code⁴ and images used in our experiments are available online at <http://rr.epfl.ch/demosaicking>.

Our main goal here is to verify that the proposed one-step implementation can indeed achieve the same results obtained by the original iterative algorithm, but at a much lower computational cost. Therefore, we choose not to report the comparison

between the AP algorithm and other demosaicking schemes in the literature, which can be found in several previous publications (e.g., [2], [3]).

In the experiments, we use the same subband filters as described in [5] (see (2) for their specifications). Correspondingly, the polyphase filters used in the proposed noniterative algorithm are precomputed as follows. Starting from the lowpass filter $L(\boldsymbol{\omega})$ defined in (5), we construct the polyphase matrix $\boldsymbol{\Phi}_L(\boldsymbol{\omega})$ and $\mathbf{T}_L(\boldsymbol{\omega})$ according to (15) and (25), respectively. The filters $F_{00}(\boldsymbol{\omega})$, $F_{10}(\boldsymbol{\omega})$, and $F_{11}(\boldsymbol{\omega})$ in Fig. 6 are then obtained by (41) (for partial convergence) and (46) (for full convergence), respectively.

Two implementation details are worth mentioning. First, the polyphase filters $F_{00}(\boldsymbol{\omega})$, $F_{10}(\boldsymbol{\omega})$, and $F_{11}(\boldsymbol{\omega})$ obtained previously can either span large spatial supports (which will depend upon the iteration number k as in the case of partial convergence) or even have infinite spatial supports [as in the case of full convergence, due to the factor $(\mathbf{I} - \mathbf{T}_L(\boldsymbol{\omega}))^{-1}$ in (46)]. In practice, however, we find that these filters can be well-approximated by their finitely-truncated versions—empirically determined to be 6×6 in our experiments. Our second observation is that, although the obtained polyphase filters are not exactly separable 2-D filters, they are close to being so. Consequently, to further improve computational efficiency, we replace all these filters with their separable approximations, constructed⁵ to be optimal in the least square sense [25].

Table I summarizes the results of the experiments, averaged over the 24 test images. For the iterative AP algorithm, we consider two different options for the number of iterations: 6 and 20. The former is preferred in practice due to its reduced computational load, whereas the latter is chosen to test the full convergence behavior. Correspondingly, we test two versions of the proposed noniterative algorithm—partial convergence and full convergence.

The performance of the algorithms is measured in terms of the PSNR of the demosaicked images. Note that the PSNR values for the green channel are the same for different algorithms, since both AP and the proposed scheme follow the same steps in estimating the missing green pixels. For the red and blue pixels, the PSNR values obtained by the proposed scheme are very close to those reached by AP. This verifies the accuracy of the pro-

⁴For the iterative AP algorithm, we use the code provided by the authors of [5].

⁵Any 2-D filter with finite support can be represented by a matrix \mathbf{A} . The optimal separable approximation of \mathbf{A} in the least square sense is $\sigma_1 \mathbf{u}_1 \otimes \mathbf{v}_1$, where σ_1 is the largest singular value of \mathbf{A} , and $\mathbf{u}_1, \mathbf{v}_1$ are the corresponding left and right singular vectors.

posed noniterative algorithm. The small differences in PSNR are caused by the finite truncation and separable approximation of the polyphase filters in our implementation.

The main advantage of the proposed noniterative scheme is its computational efficiency. We record in the table the running time of different algorithms on a computer with a 2.2 GHz CPU. In the case of partial convergence (with 6 iterations), the proposed algorithm is about 8.5 times faster than its iterative counterpart. In the case of full convergence, the speedup becomes even more significant as the number of iterations in the AP algorithm increases to 20. For a more formal representation of computational complexity, we also provide the approximate numbers of multiplications and additions required by different algorithms. In partial convergence, the proposed noniterative algorithm reduces the number of operations by factors of 5 to 7. In full convergence, the proposed algorithm only requires three separable filtering operations in the polyphase domain and, hence, the complexity reduction (more than 40-fold) is much more dramatic.

VI. CONCLUSIONS

This paper presents a detailed treatment of a classical color image demosaicking algorithm [5] based on alternating projections (AP). We provided a rigorous analysis of the convergence of the AP algorithm based upon contraction mapping, established its optimality as the unique solution to a constrained quadratic minimization problem, and proposed an efficient one-step implementation via polyphase domain filtering.

Although our focus has been exclusively on the AP algorithm, we believe that the techniques and theoretical results introduced in this work can be applied to the study and simplification of other demosaicking schemes in the literature (e.g., [9], [15], [16], [26], [27]) and to the development of new ones. It is also our hope that the present paper, serving as a showcase of the usefulness of polyphase representation, can introduce this powerful tool from multirate signal processing to the image demosaicking community.

APPENDIX

A) *The Sufficiency Part of Proposition 1:* We show that if $L(\omega) = \mathbb{1}_{\mathcal{D}}(\omega)$, then P_d is a projection operator onto the following convex set:

$$\mathcal{C} = \left\{ r : R(\omega) = \widehat{G}(\omega), \quad \text{for } \omega \in \mathcal{D}^c \right\}$$

where $\mathcal{D}^c \stackrel{\text{def}}{=} \mathbb{R}^2 \setminus \mathcal{D}$ denotes the complement of the passband support \mathcal{D} . To that end, we just need to verify, for every signal x , the following two statements: First, $y \stackrel{\text{def}}{=} P_d x \in \mathcal{C}$, and second

$$\|x - y\|^2 \leq \|x - r\|^2 \quad \text{for all } r \in \mathcal{C}. \quad (47)$$

From the definition of P_d in (4) and the condition (10), we have

$$Y(\omega) = \mathbb{1}_{\mathcal{D}}(\omega)X(\omega) + \mathbb{1}_{\mathcal{D}^c}(\omega)\widehat{G}(\omega) \quad (48)$$

and thus $y \in \mathcal{C}$. We now concentrate on showing (47). It follows from Parseval's theorem that:

$$\begin{aligned} (2\pi)^2 \|x - r\|^2 &= \int_{[-\pi, \pi]^2} |X(\omega) - R(\omega)|^2 d\omega \\ &\geq \int_{[-\pi, \pi]^2} |\mathbb{1}_{\mathcal{D}^c}(\omega) (X(\omega) - R(\omega))|^2 d\omega \\ &= \int_{[-\pi, \pi]^2} \left| \mathbb{1}_{\mathcal{D}^c}(\omega) (X(\omega) - \widehat{G}(\omega)) \right|^2 d\omega \end{aligned} \quad (49)$$

$$= \int_{[-\pi, \pi]^2} |X(\omega) - Y(\omega)|^2 d\omega. \quad (50)$$

In reaching (49) we have used the fact that $r \in \mathcal{C}$, and the equality (50) is due to (48). Applying Parseval's theorem to (50), we are done.

B) *Proof of Lemma 1:* As in (13), we can expand the image $X(\mathbf{z})$ and the filter $L(\mathbf{z})$ in terms of their polyphase components

$$\begin{aligned} X(\mathbf{z}) &= X_{00}(\mathbf{z}^2) + z_2^{-1}X_{01}(\mathbf{z}^2) + z_1^{-1}X_{10}(\mathbf{z}^2) \\ &\quad + z_1^{-1}z_2^{-1}X_{11}(\mathbf{z}^2) \\ L(\mathbf{z}) &= L_{00}(\mathbf{z}^2) + z_2^{-1}L_{01}(\mathbf{z}^2) + z_1^{-1}L_{10}(\mathbf{z}^2) \\ &\quad + z_1^{-1}z_2^{-1}L_{11}(\mathbf{z}^2). \end{aligned} \quad (51)$$

Substituting (51) into the equality $Y(\mathbf{z}) = X(\mathbf{z})L(\mathbf{z})$, we can write the product $Y(\mathbf{z})$ as a sum of 16 terms, i.e., see equation (52) at the bottom of the page. To simplify (52), we introduce four new signals $E_{00}(\mathbf{z})$, $E_{01}(\mathbf{z})$, $E_{10}(\mathbf{z})$ and $E_{11}(\mathbf{z})$, defined as

$$\begin{aligned} [E_{00}(\mathbf{z}), E_{01}(\mathbf{z}), E_{10}(\mathbf{z}), E_{11}(\mathbf{z})]^T \\ \stackrel{\text{def}}{=} \Phi_L(\mathbf{z}) [X_{00}(\mathbf{z}), X_{01}(\mathbf{z}), X_{10}(\mathbf{z}), X_{11}(\mathbf{z})]^T \\ = \Phi_L(\mathbf{z}) \xi_X(\mathbf{z}) \end{aligned} \quad (53)$$

$$\begin{aligned} Y(\mathbf{z}) &= \left(L_{00}(\mathbf{z}^2)X_{00}(\mathbf{z}^2) + z_2^{-2}L_{01}(\mathbf{z}^2)X_{01}(\mathbf{z}^2) + z_1^{-2}L_{10}(\mathbf{z}^2)X_{10}(\mathbf{z}^2) + z_1^{-2}z_2^{-2}L_{11}(\mathbf{z}^2)X_{11}(\mathbf{z}^2) \right) \\ &\quad + z_2^{-1} \left(L_{01}(\mathbf{z}^2)X_{00}(\mathbf{z}^2) + L_{00}(\mathbf{z}^2)X_{01}(\mathbf{z}^2) + z_1^{-2}L_{11}(\mathbf{z}^2)X_{10}(\mathbf{z}^2) + z_1^{-2}L_{10}(\mathbf{z}^2)X_{11}(\mathbf{z}^2) \right) \\ &\quad + z_1^{-1} \left(L_{10}(\mathbf{z}^2)X_{00}(\mathbf{z}^2) + z_2^{-2}L_{11}(\mathbf{z}^2)X_{01}(\mathbf{z}^2) + L_{00}(\mathbf{z}^2)X_{10}(\mathbf{z}^2) + z_2^{-2}L_{01}(\mathbf{z}^2)X_{11}(\mathbf{z}^2) \right) \\ &\quad + z_1^{-1}z_2^{-1} \left(L_{11}(\mathbf{z}^2)X_{00}(\mathbf{z}^2) + L_{10}(\mathbf{z}^2)X_{01}(\mathbf{z}^2) + L_{01}(\mathbf{z}^2)X_{10}(\mathbf{z}^2) + L_{00}(\mathbf{z}^2)X_{11}(\mathbf{z}^2) \right). \end{aligned} \quad (52)$$

where $\Phi_L(z)$ is the 4×4 polyphase matrix in (15), with ω , $e^{-j\omega_1}$ and $e^{-j\omega_2}$ replaced by their z -domain counterparts z , z_1^{-1} and z_2^{-1} , respectively. It is straightforward to verify that (52) can be simplified as

$$Y(z) = E_{00}(z^2) + z_2^{-1}E_{01}(z^2) + z_1^{-1}E_{10}(z^2) + z_1^{-1}z_2^{-1}E_{11}(z^2). \quad (54)$$

On the one hand, $Y(z)$ can be expanded in terms of its polyphase components as

$$Y(z) = Y_{00}(z^2) + z_2^{-1}Y_{01}(z^2) + z_1^{-1}Y_{10}(z^2) + z_1^{-1}z_2^{-1}Y_{11}(z^2).$$

On the other hand, (54) provides an alternative expansion of $Y(z)$. It follows from the uniqueness of the polyphase transform that $E_{ij}(z) = Y_{ij}(z)$ for all $i, j \in \{0, 1\}$, and therefore (53) implies (17).

C) The Lipschitz Constant of the Mapping T : We show that the Lipschitz constant α_T can be calculated as in (28). Consider two partial polyphase vectors θ_{X_1} and θ_{X_2} , and let $\theta_{Y_1} = T\theta_{X_1}$ and $\theta_{Y_2} = T\theta_{X_2}$. From Parseval's theorem

$$(2\pi)^2 \|\theta_{Y_1} - \theta_{Y_2}\|^2 = \int_{[-\pi, \pi]^2} \|\theta_{Y_1}(\omega) - \theta_{Y_2}(\omega)\|^2 d\omega. \quad (55)$$

According to the definition of the mapping T given in (25)

$$\theta_{Y_1}(\omega) - \theta_{Y_2}(\omega) = \mathbf{T}_L(\omega) (\theta_{X_1}(\omega) - \theta_{X_2}(\omega)).$$

Substitute the previously mentioned equality into (55) and write $\theta(\omega) \stackrel{\text{def}}{=} \theta_{X_1}(\omega) - \theta_{X_2}(\omega)$, we obtain equations (56)–(58) at the bottom of the page, where (56) is due to a standard inequality on matrix spectral norm [28], and (58) is from Parseval's equality. Next, we show that the inequalities in (56) and (57) are also tight. To that end, we construct

$$\theta_\gamma(\omega) = \mathbf{V}_1(\omega) \mathbb{1}_{B(\gamma)}(\omega - \omega_0) \quad (59)$$

where $\mathbf{V}_1(\omega)$ is the first right-singular vector of $\mathbf{T}_L(\omega)$, ω_0 is the frequency where $\sigma_1^2(\mathbf{T}_L(\omega))$ reaches its maximum, i.e.,

$$\omega_0 \stackrel{\text{def}}{=} \arg \max_{\omega} \sigma_1^2(\mathbf{T}(\omega))$$

and $\mathbb{1}_{B(\gamma)}(\omega - \omega_0)$ denotes the indicator function defined on a ball of radius γ , centered on ω_0 . It is easy to verify that, when we choose $\theta_\gamma(\omega)$ as in (59), the inequality (56) becomes

an equality. Furthermore, when the radius γ tends to zero, the second inequality (57) can be made arbitrarily tight.

D) The Equivalence of (29) and (30) for Symmetric Low-pass Filters: We show the equivalence of the conditions (29) and (30) for lowpass filters satisfying the following symmetric property:

$$L(z_1, z_2) = L(z_2, z_1).$$

Writing the previously mentioned equality in the polyphase domain yields

$$\begin{aligned} L(z_1, z_2) &= L_{00}(z_1^2, z_2^2) + z_2^{-1}L_{01}(z_1^2, z_2^2) \\ &\quad + z_1^{-1}L_{10}(z_1^2, z_2^2) + z_1^{-1}z_2^{-1}L_{11}(z_1^2, z_2^2) \\ &= L_{00}(z_2^2, z_1^2) + z_1^{-1}L_{01}(z_2^2, z_1^2) \\ &\quad + z_2^{-1}L_{10}(z_2^2, z_1^2) + z_1^{-1}z_2^{-1}L_{11}(z_2^2, z_1^2) \\ &= L(z_2, z_1). \end{aligned}$$

It follows from the uniqueness of the polyphase decomposition that:

$$L_{00}(z_1, z_2) = L_{00}(z_2, z_1)$$

$$L_{01}(z_1, z_2) = L_{10}(z_2, z_1)$$

and

$$L_{11}(z_1, z_2) = L_{11}(z_2, z_1).$$

From the definition of the polyphase matrix Φ_L in (15) and the constructions of \mathbf{T}_L and $\tilde{\mathbf{T}}_L$, we can verify that the previously mentioned equalities imply that

$$\mathbf{T}_L(\omega_1, \omega_2) = \tilde{\mathbf{T}}_L(\omega_2, \omega_1).$$

Consequently

$$\begin{aligned} \max_{\omega} \sigma_1 \left(\mathbf{T}_L(e^{j(\omega_1, \omega_2)}) \right) &= \max_{\omega} \sigma_1 \left(\mathbf{T}_L(e^{j(\omega_2, \omega_1)}) \right) \\ &= \max_{\omega} \sigma_1 \left(\tilde{\mathbf{T}}_L(e^{j(\omega_1, \omega_2)}) \right). \end{aligned}$$

E) Proof of Lemma 4: Since $H^*(\omega)H(\omega) = 1 - L(\omega)$, we have

$$\Phi_{H^*}(\omega)\Phi_H(\omega) = \Phi_{1-L}(\omega) = \mathbf{I} - \Phi_L(\omega).$$

What is left to show is that $\Phi_{H^*}(\omega) = \Phi_H^*(\omega)$.

$$\begin{aligned} (2\pi)^2 \|\theta_{Y_1} - \theta_{Y_2}\|^2 &= \int_{[-\pi, \pi]^2} \|\mathbf{T}_L(\omega)\theta(\omega)\|^2 d\omega \\ &\leq \int_{[-\pi, \pi]^2} \sigma_1^2(\mathbf{T}_L(\omega)) \|\theta(\omega)\|^2 d\omega \end{aligned} \quad (56)$$

$$\leq \left(\max_{\omega} \sigma_1^2(\mathbf{T}_L(\omega)) \right) \int_{[-\pi, \pi]^2} \|\theta(\omega)\|^2 d\omega \quad (57)$$

$$= \left(\max_{\omega} \sigma_1^2(\mathbf{T}_L(\omega)) \right) (2\pi)^2 \|\theta_{X_1} - \theta_{X_2}\|^2 \quad (58)$$

Let $\tilde{H}(\boldsymbol{\omega}) \stackrel{\text{def}}{=} H^*(\boldsymbol{\omega})$. In this paper, we only consider filters with real-valued coefficients, and so, in the z -domain, $\tilde{H}(z) = H(z^{-1})$. Expanding $H(z)$ in terms of its polyphase components as in (13) and replacing z with z^{-1}

$$\begin{aligned} \tilde{H}(z) &= H(z^{-1}) \\ &= H_{00}(z^{-2}) + z_2^{-1} (z_2^2 H_{01}(z^{-2})) \\ &\quad + z_1^{-1} (z_1^2 H_{10}(z^{-2})) \\ &\quad + z_1^{-1} z_2^{-1} (z_1^2 z_2^2 H_{11}(z^{-2})). \end{aligned} \quad (60)$$

Due to the uniqueness of the polyphase expansion, we can determine from (60) the polyphase components of $\tilde{H}(z)$ as $\tilde{H}_{00}(z) = H_{00}(z^{-1})$, $\tilde{H}_{01}(z) = z_2 H_{01}(z^{-1})$, $\tilde{H}_{10}(z) = z_1 H_{10}(z^{-1})$, and $\tilde{H}_{11}(z) = z_1 z_2 H_{11}(z^{-1})$. Going back to the Fourier domain and using the definition of the polyphase matrix in (15), we can easily verify that $\Phi_{\tilde{H}}^*(\boldsymbol{\omega}) = \Phi_H^*(\boldsymbol{\omega})$.

REFERENCES

[1] R. Ramanath, W. E. Snyder, G. L. Bilbro, and W. A. Sander, III, "Demosaicking methods for Bayer color arrays," *J. Electron. Imag.*, vol. 11, no. 3, pp. 306–315, Jul. 2002.

[2] 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.

[3] X. Li, B. Gunturk, and L. Zhang, W. A. Pearlman, J. W. Woods, and L. Lu, Eds., "Image demosaicing: A systematic survey," in *Proc. SPIE Conf. on Vis. Commun. and Image Proc.*, San Jose, CA, Jan. 2008, vol. 6822.

[4] D. Alleysson and B. C. de Lavarène, W. A. Pearlman, J. W. Woods, and L. Lu, Eds., "Frequency selection demosaicking: A review and a look ahead," in *Proc. SPIE Conf. on Vis. Commun. and Image Proc.*, San Jose, CA, Jan. 2008, vol. 6822.

[5] 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.

[6] R. Lukac, K. Martin, and K. N. Plataniotis, "Demosaicked image post-processing using local color ratios," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 6, pp. 914–920, Jun. 2004.

[7] X. Wu and N. Zhang, "Primary-consistent soft-decision color demosaicking for digital cameras," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1263–1274, Sep. 2004.

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

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

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

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

[12] 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. and Technol.*, vol. 17, no. 3, pp. 105–122, Oct. 2007.

[13] 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.

[14] D. G. Luenberger, *Optimization by Vector Space Methods*. Hoboken, NJ: Wiley, 1969.

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

[16] L. Condat, "A generic variational approach for demosaicking from an arbitrary color filter array," in *Proc. IEEE Int. Conf. on Image Proc.*, Cairo, Egypt, Nov. 2009.

[17] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Upper Saddle River, NJ: Prentice-Hall, 1993.

[18] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Upper Saddle River, NJ: Prentice Hall, 1995.

[19] B. E. Bayer, "Color Imaging Array," U.S. 3 971 065, Jul. 1976.

[20] J. F. Hamilton and J. E. Adams, "Adaptive Color Plane Interpolation in Single Sensor Color Electronic Camera," U.S. 5 629 734, May 1997.

[21] J. E. Adams, "Design of practical color filter array interpolation algorithms for digital cameras, part 2," in *Proc. IEEE Int. Conf. on Image Proc.*, Chicago, IL, Oct. 1998.

[22] W. Cheney and A. A. Goldstein, "Proximity maps for convex sets," *Proc. Amer. Math. Soc.*, vol. 10, no. 3, pp. 448–450, Jun. 1959.

[23] A. W. Naylor and G. R. Sell, *Linear Operator Theory in Engineering and Science*. New York: Springer-Verlag, 1982.

[24] D. H. Brandwood, "A complex gradient operator and its applications in adaptive array theory," *IEE Proc. F, Commun., Radar, Signal Process.*, vol. 130, no. 1, pp. 11–16, Feb. 1983.

[25] W.-S. Lu and A. Antoniou, *Two-Dimensional Digital Filters*. Boca Raton, FL: CRC, 1992.

[26] H. J. Trussel and R. E. Hartwig, "Mathematics for demosaicking," *IEEE Trans. Image Process.*, vol. 11, no. 4, pp. 485–492, Apr. 2002.

[27] B. C. de Lavarène, D. Alleysson, and J. Héraud, "Practical implementation of LMMSE demosaicing using luminance and chrominance spaces," *Comput. Vis. Image Understand.*, vol. 107, no. 1-2, pp. 3–13, 2007.

[28] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge University Press, 1985.



Yue M. Lu (S'04–M'07) received the B.Eng and M.Eng degrees in electrical engineering from Shanghai Jiao Tong University, China, in 1999 and 2002, respectively. He received the M.Sc. degree in mathematics and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign in 2007.

He was a Research Assistant at the University of Illinois at Urbana-Champaign, and has worked for Microsoft Research Asia, Beijing, China and Siemens Corporate Research, Princeton, NJ. In September 2007, He joined the Audiovisual Communications Laboratory at Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, where he was a postdoctoral researcher and lecturer. Since 2010, he has been an Assistant Professor of Electrical Engineering at Harvard University, Cambridge, MA. His research interests include signal processing for sensor networks; the theory, constructions, and applications of multiscale representations for multidimensional signals; image and video processing; and sampling theory.

Dr. Lu received the Most Innovative Paper Award of IEEE International Conference on Image Processing (ICIP) in 2006 for his paper (with Minh N. Do) on the construction of directional multiresolution image representations, and the Student Paper Award of IEEE ICIP in 2007.



Mina Karzand received the B.S. degree in electrical and electronics engineering from the University of Tehran, Tehran, Iran, in 2007, the M.S. degree in electrical engineering from Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, in 2009, and is currently pursuing the Ph.D. degree at the Research Laboratory of Electronics (RLE), Massachusetts Institute of Technology (MIT), Cambridge.



Martin Vetterli (F'95) received the Dipl. El.-Ing. degree from ETH Zurich (ETHZ), Switzerland, in 1981, the M.S. degree from Stanford University, Stanford, CA, in 1982, and the Doctorat ès Sciences degree from EPF Lausanne (EPFL), Switzerland, in 1986.

He was a Research Assistant at Stanford University and EPFL, and has worked for Siemens and AT&T Bell Laboratories. In 1986, he joined Columbia University, New York, where he was last an Associate Professor of Electrical Engineering and codirector of the Image and Advanced Television Laboratory. In 1993, he joined the University of California at Berkeley, where he was a Professor with the Department of Electrical Engineering and Computer Sciences until 1997, and now holds an Adjunct Professor position. Since 1995, he has been a Professor of Communication Systems at EPF Lausanne, Switzerland, where he chaired the Communications Systems Division (1996/1997), and heads the Audiovisual Communications Laboratory. From 2001 to 2004, he directed the National Competence Center in Research on mobile information and communication systems. He is also a Vice-President for International Affairs at EPFL since October 2004. He has held visiting positions with ETHZ (1990) and Stanford University (1998). His research interests include sampling, wavelets, multirate signal processing, computational complexity, signal processing for communications, digital image/video processing, joint source/channel coding, and signal processing for sensor networks. He is the coauthor of books with J. Kovačević (*Wavelets and Subband Coding*), with P. Prandoni (*Signal Processing for Communications*), and with J. Kovačević and V. K. Goyal (*Fourier and Wavelet Signal Processing*). He has published about 120 journal papers on a variety of topics in signal/image processing and communications, holds a dozen patents, and is an ISI highly cited researcher in engineering.

Dr. Vetterli is a fellow of the IEEE, a fellow of the ACM, and a member of SIAM. He is on the editorial boards of *Applied and Computational Harmonic Analysis*, *The Journal of Fourier Analysis and Application*, and the IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING. He received the Best Paper Award of EURASIP in 1984 for his paper on multidimensional subband coding, the Research Prize of the Brown Boverly Corporation (Switzerland) in 1986 for his doctoral thesis, the IEEE Signal Processing Society's Senior Paper Awards in 1991, 1996, and 2007 (for papers with D. LeGall and K. Ramchandran, and Marziliano and Blu, respectively). He won the Swiss National Latsis Prize in 1996, the SPIE Presidential award in 1999, and the IEEE Signal Processing Technical Achievement Award in 2001. He was a member of the Swiss Council on Science and Technology until December 2003. He was a plenary speaker at various conferences (e.g., IEEE ICIP, ICASSP, ISIT).