Demosaicing by Successive Approximation

Xin Li, *Member, IEEE*

***Abstract*—In this paper, we present a fast and high-performance algorithm for color filter array (CFA) demosaicing. CFA demosaicing is formulated as a problem of reconstructing correlated signals from their downsampled versions with an opposite phase. The major contributions of this work include 1) a new iterative demosaicing algorithm in the color difference domain and 2) a spatially adaptive stopping criterion for suppressing color misregistration and zipper artifacts in the demosaiced images. We have compared the proposed demosaicing algorithm with two current state-of-the-art techniques reported in the literature. Ours outperforms both of them on demosaicing performance and computational cost.**

***Index Terms*—Bayer pattern, color difference rule, color filter array (CFA), color misregistration artifacts, color ratio rule, iterative demosaicing, successive approximation.**

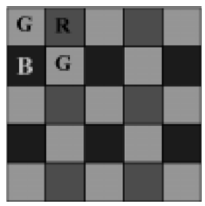

Fig. 1. Bayer pattern (U.S. patent 3 971 065).

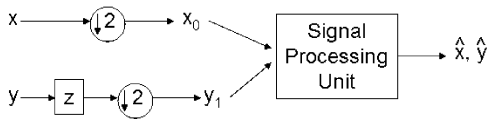

Fig. 2. Problem of reconstructing correlated signals from their decimated versions.

I. Introduction

DEMOSAICING, also called color filter array (CFA) interpolation, refers to the problem of reconstructing a color image from the charge-coupled device (CCD) samples. The most commonly used CCD samples are Bayer pattern, as shown in Fig. 1. In Bayer pattern, the sampling density of green pixels is twice as much as that of red or blue pixels. The objective of demosaicing is to interpolate the missing red, green and blue pixels from the available ones so that the reconstructed image can be as close to the original full-resolution color image as possible. Meanwhile, the computational complexity of demosaicing needs to be kept low for the reason of being cost effective in practical applications.

Like other color image processing problems, modeling the correlation among three color channels (planes) plays the critical role in demosaicing. Roughly speaking, all color channels have very similar characteristics such as texture and edge location. Ignoring such interplane dependency (e.g., straightforward intraplane linear interpolation [5]) often renders the demosaiced image suffering from annoying artifacts caused by color misregistration. Various techniques have been proposed to obtain a more faithful and higher quality reproduction of color images by exploiting the interplane correlation. The grand challenge is to find the best tradeoff between image quality and computational cost.

We classify previous demosaicing techniques into two categories: *noniterative* [6]–[15] and *iterative* [16], [17]. Noniterative demosaicing techniques mainly rely on the idea of edge-directed interpolation to improve the reconstruction performance. The exploitation of intraplane correlation can be done by estimating either local gradients [3], [4] or local covariance information [14]. The exploitation of interplane correlation can be performed based on either the color ratio rule [16] or the color difference rule [7]. Iterative demosaicing techniques have also been proposed recently [16], [17]. In [16], green and red/blue channels are iteratively updated by enforcing the color ratio rule. In [17], a projection-onto-convex-set (POCS)-based technique is proposed to refine the red and blue planes by alternatively enforcing the two convex-set constraints. It has been observed that iterative demosaicing techniques are capable of achieving higher quality in the reconstructed images than noniterative ones at the price of increased computational cost.

In this paper, we formulate CFA demosaicing as a problem of reconstructing correlated signals (say x and y) from their downsampled samples with opposite phases, i.e., x_0, y_1 (subscripts 0, 1 denote the *phase* attribute), as shown in Fig. 2. Such a problem widely exists in the field of image and video processing. For example, in the problem of deinterlacing [20], x and y are adjacent video frames and x_0, y_1 correspond to even field and odd field, respectively; the correlation between x and y is characterized by the motion model. In the problem of CFA demosaicing, x, y are green plane and red/blue planes and x_0 and y_1 are samples at two quincunx lattices respectively; the correlation between x and y is characterized by the interplane dependency.

A common observation with such types of problems is the "chicken-and-egg" flavor, i.e., in order to recover x_1 (the missing phase in signal x), we attempt to exploit the correlation between x and y, which requires the knowledge of y_0 (the missing phase in signal y) and vice versa. Such observation motivates us to come up with a novel iterative formulation—starting from a rough guess, we can alternatively refine our estimate of x_1, y_0 (such an idea is elaborated for

Manuscript received July 24, 2003; revised January 20, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yucel Altunbasak.

The author is with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26506 USA (e-mail: xinl@csee.wvu.edu).

Digital Object Identifier 10.1109/TIP.2004.840683

the demosaicing problem in Section III-B). The fundamental assumption behind is that the correlation model with x and y is sufficiently accurate such that an improved estimate of the missing phase in one signal must lead to an improved estimate of the missing phase in the other.

There are two popular interplane correlation models in the literature: the color *ratio* rule [16] and the color *difference* rule [7]. We advocate the use of the color difference rule because it has lower computational cost and better fits the linear interpolation models widely used in the literature of CFA demosaicing. Another significant contribution of this paper is to study the cause of notorious color misregistration and zipper artifacts with demosaiced images and propose an efficient strategy of suppressing them. It is shown that color misregistration artifacts are closely related to frequency aliasing in CFA and zipper artifacts are caused by over-enforcing the color difference rule in iterative demosaicing. A spatially adaptive stopping criterion is proposed to effectively suppress both types of artifacts. We compare the proposed demosaicing technique and two state-of-the-art algorithms [17], [8] for a collection of 12 photographic color images. Both objective and subjective evaluation of image quality support the effectiveness of our scheme. The computational cost of our new iterative demosaicing technique also appears to be lower than that of existing ones.

The rest of this paper is organized as follows. Section II studies two approaches of modeling the interplane correlation in color images: the color ratio rule and the color difference rule. Section III presents the new iterative demosaicing algorithm and studies its initial condition as well as its convergence property. Section IV investigates the cause of color misregistration and zipper artifacts and proposes a spatially adaptive stopping criterion to suppress both types of artifacts. Experiment results are reported in Section V and concluding remarks are made in Section VI.

II. Modeling Color Interplane Correlation

In this section, we study two approaches of exploiting interplane correlation among different color channels: the color ratio rule [16] and the color difference rule [10], [7]. Both rules represent heuristic approximation of nonlinear dependency among color planes. It has been found [2], [3] that the color difference rule is more suitable for CFA data after gamma correction than the color ratio rule. Our contribution is to demonstrate that when interplane correlation is combined with intraplane correlation (often characterized by a linear interpolative model), the color difference rule works more effectively (faster and more stable), especially for the class of iterative CFA demosaicing schemes.

A. Color Ratio Rule

The color ratio rule employed in [16] is based on a simplified approach of the modeling color image formation by viewing Lambertian nonflat surface patches. According to the model, the pixel intensity of each channel may be viewed as the projection of the surface normal $\vec{N}(x)$ onto the light source $\vec{l}$, multiplied by the albedo ρ. The albedo ρ is a parameter of the surface material and differs for varying spectral channels. If we use $\rho_a (a = R, G, B)$ to denote the albedo for each channel, the three color planes can be represented by [16]

$$I_a(x) = \rho_a(x)\langle \vec{N}(x), \vec{l} \rangle \tag{1}$$

where $I_a(x)$ is the pixel intensity of channel a at the location of x. It is often assumed that within a given object of an image, the albedo is constant and, therefore, the following color ratio rule is valid within the interior of the given object [16] (superscripts a, b denote the color plane index):

$$K_{a,b}(x) = \frac{I_a(x)}{I_b(x)} = \frac{\rho_a}{\rho_b} = \text{constant}. \tag{2}$$

Although the above rule is based on an oversimplified assumption of the color imaging model, it has shown to be fairly effective on exploiting the interplane correlation in demosaicing applications [16].

B. Color Difference Rule

Another simplified way of modeling interplane correlation is to assume that the color difference signal is approximately flat within the boundary of a given object [3], [7]. That is

$$D_{a,b}(x) = I_a(x) - I_b(x) = \text{constant}. \tag{3}$$

Apparently, such a color difference rule enjoys computational efficiency over the color ratio rule due to the lack of nonlinear division operations.

The color ratio rule and the color difference rule can be unified at least theoretically by a nonlinear warping (gamma correction) of image intensity values, i.e., the color difference rule defined with respect to $\log(I(x))$ is equivalent to the color ratio rule defined with respect to $I(x)$. However, which rule is more appropriate for the demosaicing application depends on the way we exploit both *intra* and *inter*plane correlation in practice. For example, it is often for the reason of exploiting *intra*plane correlation and computational efficiency that the missing color ratio or difference is interpolated by a linear combination of its neighboring values. Therefore, if we think of $K_{a,b}(x)$, $D_{a,b}(x)$ as two stochastic processes, the question of choosing *ratio* or *difference* boils down to which process better fits the assumed linear interpolation model. It is from such model fitting (and also computational cost) perspective that we argue that the color difference rule has certain advantages over the color ratio rule.

C. Justifications

To justify the above claim, let us consider a simplified model for $I(x) : I(x+\tau) = \tau I(x+1) + (1-\tau)I(x)$ (i.e., $I(x)$ satisfies the bilinear interpolative model). With some algebraic manipulation, we can easily verify that such model still holds in the domain of the color difference

$$D_{a,b}(x+\tau) = \tau D_{a,b}(x+1) + (1-\tau)D_{a,b}(x). \tag{4}$$

However, due to the curse of nonlinearity, the linear interpolative model usually would not be valid in the domain of the color ratio, i.e.

$$K_{a,b}(x+\tau) \neq \tau K_{a,b}(x+1) + (1-\tau)K_{a,b}(x). \tag{5}$$

TABLE I
INTERPOLATIVE GAIN (IN DECIBELS) COMPARISON BETWEEN COLOR RATIO AND COLOR DIFFERENCE FOR THE COLLECTION OF 12 KODAK TEST IMAGES

image	1	2	3	4	5	6	7	8	9	10	11	12
$G(K_{R,G})$	23.53	12.72	20.69	9.39	16.05	8.99	15.97	8.83	2.55	14.87	6.86	11.91
$G(D_{R,G})$	31.05	24.34	27.53	22.94	23.78	27.25	20.64	23.40	19.87	26.66	22.42	14.43
$G(K_{B,G})$	24.71	9.59	17.36	9.65	20.05	11.43	16.19	16.19	15.35	16.17	18.92	12.95
$G(D_{B,G})$	30.45	22.94	23.88	22.15	25.94	20.81	24.53	25.69	23.58	25.30	25.44	15.24

Fig. 3. Collection of 12 test images used in the experiments (images are numbered from 1 to 12 in the order of left-to-right and top-to-bottom).

In fact, it is not difficult to find a counterexample in which the error of linear interpolative model can be made arbitrarily large, which makes the set defined by the color ratio rule nonconvex. We believe that such lack of convexity at least partially contribute to the lack of convergence of the color ratio rule-based iterative demosaicing such as [16].

We have also empirically compared the color ratio and the color difference rules. To qualitatively measure the appropriateness of linear interpolation models for a stochastic process X, the following "interpolative gain" (an analogy to the conventional "predictive gain" [21]) is calculated from the image data

$$G(X) = 10\log_{10}\frac{Var(X)}{Var(e)} \tag{6}$$

where X is either $K_{a,b}(x)$ or $D_{a,b}(x)$ and e is the residue signal filtered by a Laplacian operator

$$\frac{1}{4}\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}. \tag{7}$$

We have computed the interpolative gains of $G(K_{a,b}), G(D_{a,b})$ for the set of Kodak photographic color image database (40 images in total). It has been found that the values of $G(D_{a,b})$ are uniformly larger than $G(K_{a,b})$ for each pair of (a,b) and every image. Due to the space limitation, we only report our findings (refer to Table I) with a subset of 12 images which have also been used in previous work on demosaicing [17]. These 12 images (as shown in Fig. 3) are selected because they cover a wide range of texture and color patterns in the real world.

III. ITERATIVE DEMOSAICING ALGORITHM

In this section, we present a fast and high-performance iterative demosaicing algorithm. The low computational complexity comes from color-difference interpolation, which does not involve nonlinear [16] or filtering [17] operations. The high performance comes from intelligent exploitation of color correlation in a successive approximation framework.

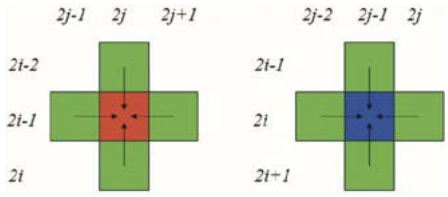

Fig. 4. Updating green channel. (Left) Interpolate D_R at the location of red sublattice. (Right) Interpolate D_B at the location of blue sublattice.

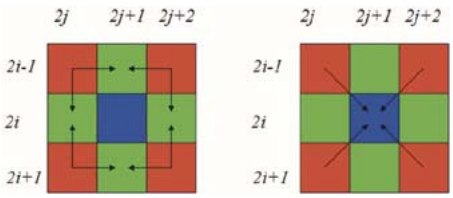

Fig. 5. Updating red channel. (Left) Interpolate D_R at the location of green sublattice. (Right) Interpolate D_R at the location of blue sublattice.

A. Color-Difference Interpolation

From now on, we use $x = (i,j)$ ($1 \le i \le H, 1 \le j \le W$, H, W are height and width of the image) to denote the spatial coordinate and superscript n to denote the iteration index. Our starting point is a coarse approximation of color planes (R^0, G^0, and B^0) obtained by intraplane interpolation techniques (e.g., bilinear [5], edge-sensitive interpolation [4], or edge directed [14]). Such coarse approximation ignores the interplane correlation and therefore violates the color difference rule. To enforce the color difference rule across different color channels, it has been suggested [3], [7] that linear interpolation be applied with the color difference instead of the original color planes. For the reason of completeness, we summarize the basic ideas behind [7] as follows.

Since green plane is the dominant channel (affects the luminance mostly), it is used as the reference and the following two color difference signals (chrominance)

$$D_R(i,j) = R(i,j) - G(i,j), D_B(i,j) = B(i,j) - G(i,j) \tag{8}$$

work as the bridge to pass on successively refined estimate of missing data between green and red/blue channels.

The green pixel at the location of red sublattice (i = odd, j = even) is updated by (see Fig. 4)

$$\hat{D}_R^n(2i-1,2j) = \frac{1}{4}[D_R^n(2i-2,2j) + D_R^n(2i,2j) + D_R^n(2i-1,2j-1) + D_R^n(2i-1,2j+1)] \tag{9}$$

and

$$G^{n+1}(2i-1,2j) = R^n(2i-1,2j) - \hat{D}_R^n(2i-1,2j). \tag{10}$$

The processing of blue sublattice (i = even and j = odd) is similar.

The red pixels are updated in two steps (see Fig. 5).

1) Interpolate the red pixels at $i + j$ = even, i.e.

$$\hat{D}_R^n(2i,2j) = \frac{1}{2}[D_R^n(2i-1,2j) + D_R^n(2i+1,2j)] \tag{11}$$

$$\hat{D}_R^n(2i+1,2j+1) = \frac{1}{2}[D_R^n(2i+1,2j) + D_R^n(2i+1,2j+2)] \tag{12}$$

and

$$R^{n+1}(2i,2j) = G^n(2i,2j) + \hat{D}_R^n(2i,2j) \tag{13}$$

$$R^{n+1}(2i+1,2j+1) = G^n(2i+1,2j+1) + \hat{D}_R^n(2i+1,2j+1). \tag{14}$$

2) Interpolate the red pixels at i = even, j = odd, i.e.

$$\hat{D}_R^n(2i,2j-1) = \frac{1}{4}[D_R^n(2i-1,2j-1) + D_R^n(2i+1,2j-1) + D_R^n(2i,2j-2) + D_R^n(2i,2j)] \tag{15}$$

and

$$R^{n+1}(2i,2j-1) = G^n(2i,2j-1) + \hat{D}_R^n(2i,2j-1). \tag{16}$$

The treatment of blue pixels is similar to that of red pixels.

The above updating strategies are chosen for their low computational cost. As we will see next, by extending the color difference interpolation into an iterative framework, we can achieve significant improvement on demosaicing performance at the price of modestly increased complexity.

B. Successive Approximation Strategy

As mentioned in the introduction, CFA demosaicing represents a class of problems with the "chicken-and-egg" flavor, i.e., the refinement of green pixels at $i+j$ = odd and red/blue pixels at the $i+j$ = even are mutually dependent and jointly beneficial to each other. It is natural to introduce an iterative strategy to handle such type of problem, as shown in Fig. 6. The philosophy behind iterative demosaicing is simple: since an improved estimate of green pixels could lead to an improved estimate of red/blue pixels and vice versa, why don't we feed the improved estimate of R, G, and B back to the loop and hope for further improvement? By forming a closed loop, it becomes possible to successively approximate the signal with the missing phase. Such observation is the key to the success of iterative demosaicing techniques proposed in this paper. Under the iterative demosaicing framework, there are three important questions to answer: 1) Where to start? 2) Does it converge? 3) Where to stop?

First, the performance of our iterative demosaicing algorithm indeed depends on the starting point (the intraplane interpolation method for green pixels). This is not surprising because the color difference rule only emphasizes the correlation between color planes and ignores the geometric constraint of edges [14] (intraplane correlation). The only place to incorporate our *a priori* knowledge about edges is the initialization stage. As we will see in the experiment results, a sophisticated intraplane interpolation strategy for green pixels often renders better performance than ad-hoc bilinear strategy. In order to keep the overall computational complexity low, we suggest the use of edge-sensitive interpolation [4], which has also been adopted by [17].

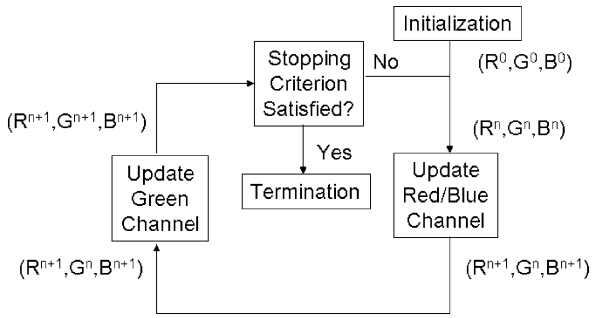

Fig. 6. Diagram of the proposed iterative demosaicing algorithm.

Second, the convergence property is critical to our understanding of iterative demosaicing algorithms. For example, in the color ratio rule-based iterative demosaicing [16], the algorithm is artificially terminated after three iterations due to lack of convergence. In fact, we can use the theory of projection-onto-convex-set (POCS) to show that the proposed iterative algorithm in Fig. 6 does converge (see Appendix). The proof is conceptually similar to that provided in [17] except that the detail constraint set defined in [17] is nonlinear while the constraint set based on the color difference rule in our scheme is linear. Experimental results in Section V also support the convergence of the proposed iterative algorithm.

Unfortunately, despite the convergence property of our iterative demosaicing algorithm, the limiting solution is *not* optimal because the color difference rule only represents an approximation of the fundamental law of color images. It has been widely observed that neither the color difference rule nor the color ratio rule is accurate enough to characterize the interplane correlation across the boundary of different objects [16], [7]. As we will see next, the limiting solution often suffers from annoying zipper artifacts and is not the right target to pursue. Instead, we propose to terminate the iteration when a carefully chosen stopping criterion is satisfied. Such characteristics, which distinguish this work from other iterative demosaicing techniques (e.g., alternating projection that does target at the limiting solution [17]), gives the name "demosaicing by successive approximation."

IV. Stopping Criterion for Artifacts Suppression

In this section, we first study the cause of two plagues with CFA demosaicing: color misregistration and zipper artifacts. Then, we propose a spatially adaptive stopping criterion for suppressing both types of artifacts.

A. Color Misregistration and Zipper Artifacts

Our studies with the proposed iterative demosaicing technique indicate that there are mainly two types of sources contributing to artifacts in a demosaiced image: 1) when the color difference rule is not sufficiently enforced for certain regions, we often observe annoying color misregistration artifacts (see Fig. 7); 2) when the color difference rule is overly enforced for certain regions, we might observe zipper artifacts [8] (see

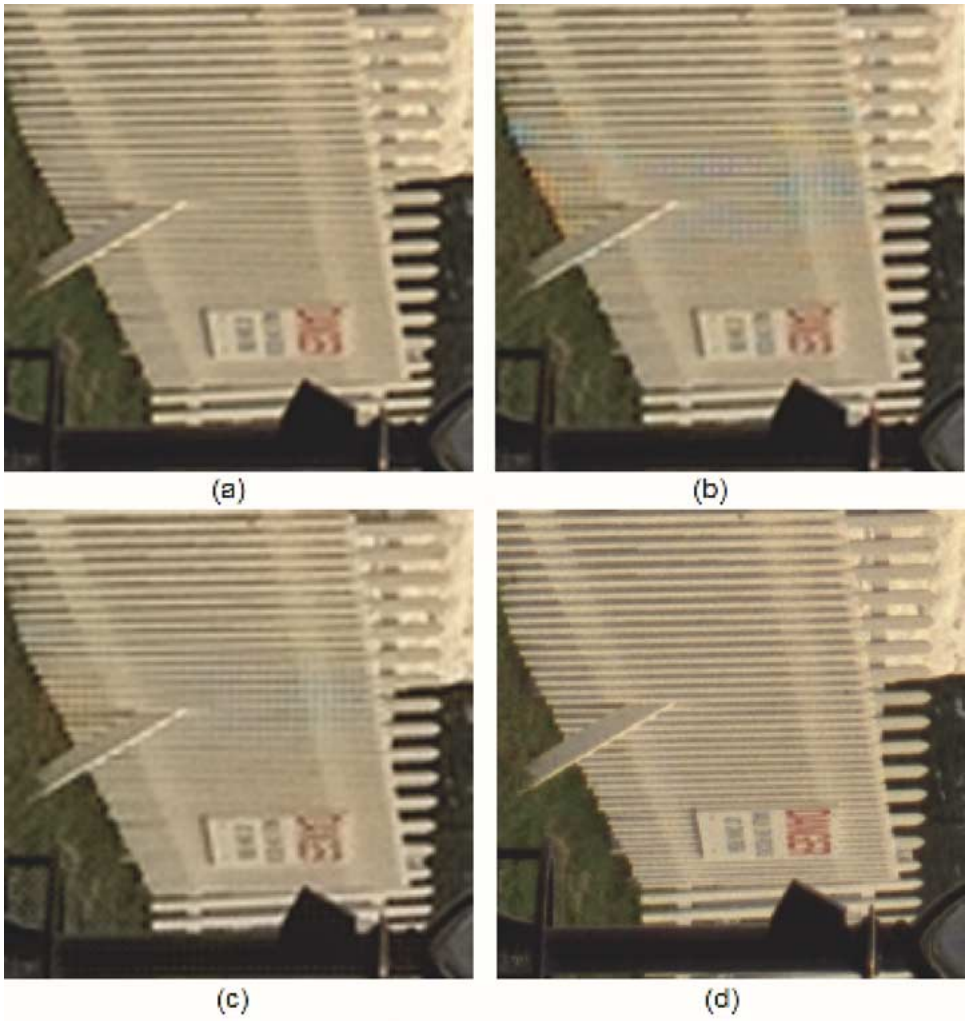

Fig. 7. (a) Original portion (100 × 100) of *img*8 at the low resolution ($s = 2$). (b) Demosaiced 100 × 100 image after three iterations. (c) Demosaiced 100 × 100 image after 20 iterations. (d) Demosaiced 400 × 400 image of the same portion after three iterations at the full resolution ($s = 0$).

Fig. 8). To the best of our knowledge, the cause of these artifacts has not been fully understood in the open literature.

In this paper, we argue that color misregistration artifacts are closely related to *frequency aliasing* in the CFA image. It is trivial to observe that the amount of aliasing in green plane is less than that in red/blue planes due to a higher sampling ratio. The nontrivial part, which has been largely overlooked by previous work on iterative demosaicing, is that the impact of frequency aliasing on image demosaicing is a complicated spatially varying phenomenon. Roughly speaking, aliasing renders it more difficult to enforce the color difference rule for spatially high-frequency components (edge areas) than spatially low-frequency components (smooth regions). When the sampling distance of Bayer pattern happens to match the interedge distance (hence, serious aliasing is introduced), it requires more iterations to enforce the color difference rule. The notorious color misregistration artifacts is the direct consequence of insufficient number of iterations.

To observe the impact of spatial frequency and aliasing on CFA demosaicing, we take a typical image from our data set (*img*8) as the example. The *fence* portion of this image has small interedge distance and consequently suffers from serious aliasing as spatial resolution drops to 512 × 768. Fig. 7(b)–(d) compare the demosaicing result of the fence portion at different resolutions. It can be observed that at the resolution of 512 × 768, it takes as many as 20 iterations to get away with the color misregistration artifacts [compare Fig. 7(b) and (c)]. While at the full resolution (2048 × 3072, no aliasing), iterative demosaicing achieves satisfactory result after just three iterations [Fig. 7(d)]. By contrast, if we inspect the other portion of 512 × 768 *img*8 which contains less aliasing (refer to Fig. 8), it also takes as few as three iterations to achieve satisfactory results.

Unlike color misregistration artifacts, zipper artifacts represent the other end of the spectrum—they often show up when we terminate the iterations not too early but too late (of course, zipper artifacts could also occur if no iteration is performed at all, but here our focus is iterative demosaicing). It is easy to see that a worse estimate of green pixels will also lead to a worse estimate of red/blue pixels (i.e., iterative demosaicing enters the negative-gain loop). Since the color difference rule only approximately holds within the object boundary, over-enforcing it has the risk of violating geometric constraint of edges and introducing zipper artifacts. This explains why zipper ar-

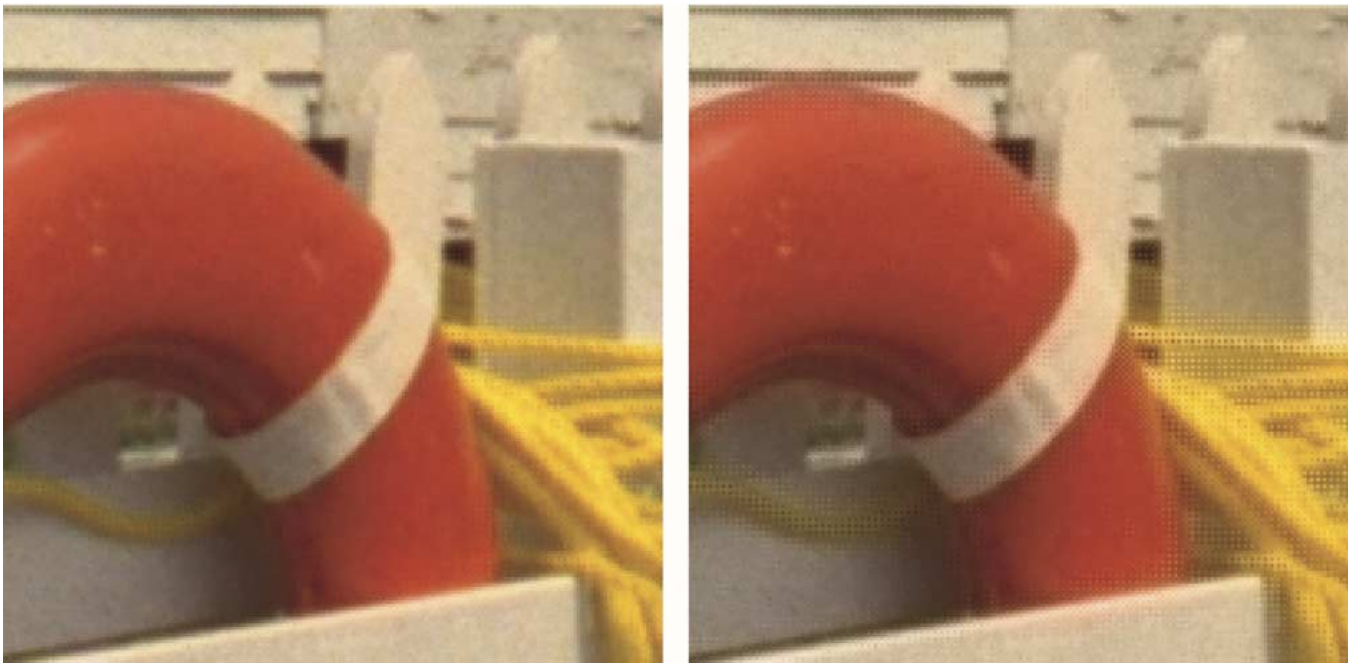

Fig. 8. Limiting solution of iterative demosaicing is risky to zipper artifacts. (Left) Demosaiced image after three iterations. (Right) Demosaiced image after 20 iterations (limiting solution).

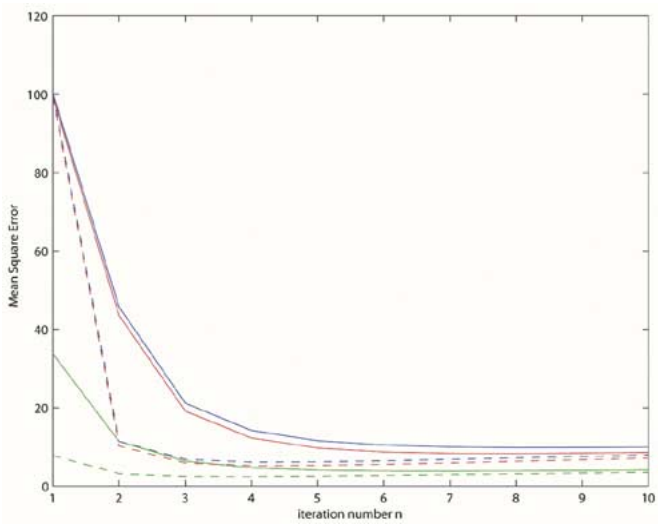

Fig. 9. Impact of starting condition on MSE performance for *img*8. Solid line: bilinear interpolation. Dashed line: edge-sensitive interpolation.

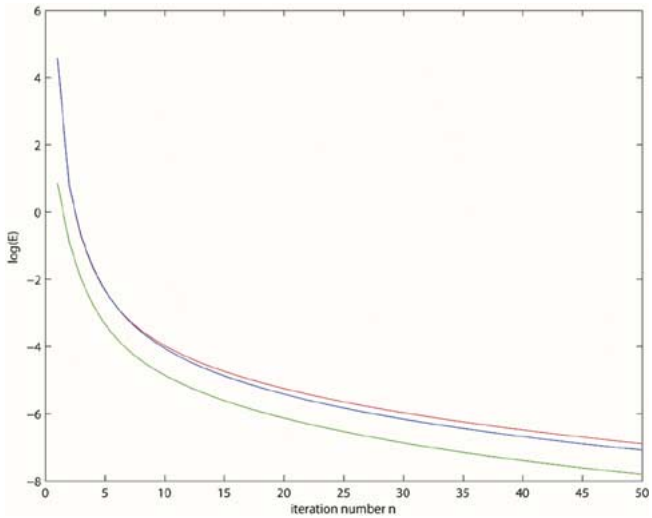

Fig. 10. Evolution of $\log(E_r)$, $\log(E_g)$, and $\log(E_b)$ as the iteration number increases.

tifacts mostly show up around smooth regions near edges. For example, Fig. 8 shows the demosaicing result for the *tyre* portion of *img*8 at the resolution of 512×768, where low spatial-frequency components are dominant and aliasing is barely noticeable. Iterative demosaicing achieves satisfactory results after three iterations; however, if we let iterative demosaicing run until the convergence, undesirable zipper artifacts stick out.

B. Spatially Adaptive Stopping Criterion

Based on the above discussion, we conclude that iterative demosaicing requires a spatially adaptive stopping criterion in order to minimize the risk of introducing either type of artifacts. For certain regions Ω_l (low aliasing), we want to select a large threshold to terminate the algorithm after few iterations; but for other regions Ω_h (high aliasing), we want to choose a small threshold so the iteration will continue until close to the convergence. The key is how to design an appropriate strategy of separating Ω_l from Ω_h.

To match the linear interpolative models of color difference in our iterative demosaicing algorithm, we propose the following classification strategy for a given image.

1) Calculate the color difference signals $D_R = R - G$, $D_B = B - G$.
2) Apply Laplacian filter defined in **(7)** to D_R, D_B to obtain e_R, e_B.
3) Declare pixel $(i,j) \in \Omega_h$ if $|e_R(i,j)| > th$ or $|e_B(i,j)| > th$; otherwise, $(i,j) \in \Omega_l$.

With the classification result, the iteration at the position (i,j) is terminated if the following condition is satisfied

$$\begin{aligned} E_r &= \|R^{n+1}(i,j) - R^n(i,j)\|_2 < \delta \\ E_g &= \|G^{n+1}(i,j) - G^n(i,j)\|_2 < \delta \\ E_b &= \|B^{n+1}(i,j) - B^n(i,j)\|_2 < \delta \end{aligned} \tag{17}$$

where the threshold is given by

$$\delta = \begin{cases} \delta_l & (i,j) \in \Omega_l \\ \delta_h & (i,j) \in \Omega_h \end{cases}. \tag{18}$$

It should be noted that the stopping criterion is based on the mean-square error (MSE) of color planes between the nth and the $n+1$th iteration. Due to the convergence of our iterative demosaicing algorithm, the values of E_r, E_g, and E_b would monotonically decrease to zero (refer to Fig. 10), which supports the validity of threshold-based stopping criterion. The suggested threshold values are $\delta_h = 0.05$, $\delta_l = 4.0$. As we can see in the next section, the proposed spatially adaptive stopping criterion is highly useful to the suppression of both color misregistration and zipper artifacts.

We summarize our iterative demosaicing algorithm as follows.

Demosaicing Algorithm by Successive Approximation:

1) Initialization: Use standard intraplane interpolation techniques (bilinear **[5]**) to fill in the missing R and B pixels and edge-sensitive interpolation **[4]** to handle G pixels.
2) Spatial classification: Use the classification strategy proposed above to divide Ω into two regions: $\Omega = \Omega_h \cup \Omega_l$.
3) Iteration: Alternate the following two updating procedures.
Update the R channel by **(11)**-**(16)** and the B channel similarly.
Update the G channel at red pixels by **(9)**-**(10)** and similarly at blue pixels.
4. Termination: Stop the iteration as soon as the stopping criterion **(17)**-**(18)** is satisfied.

Remark: If it is known *a priori* that the aliasing contained in CFA is negligible (e.g., when we apply iterative demosaicing at the full resolution 2048×3072), we might skip step 2) and directly label all pixels as low-aliasing class (i.e., $\Omega_l = \Omega$, $\Omega_h = \phi$).

V. Simulation Results

In this section, we report our experiment results with a subset of 12 Kodak photographic images. Kodak test images are available at a variety of resolutions ($2048/2^s \times 3072/2^s$, $s = 0-5$). Most previous works on image demosaicing are reported for $s = 2$ (quarter-resolution, noticeable aliasing). The software implementation of our demosaicing algorithm (MATLAB source codes) along with the 12 512×768 test images are available at http://www.csee.wvu.edu/~xinl/demo/demosaicing.html.

TABLE II
Mean Square Error Performance Comparison of Different Demosaicing Methods

Image	Channel	Scheme [7]	Scheme [8]	Scheme [17]	This work
$img1$	R	8.28	3.26	**2.74**	4.62
	G	2.34	**1.51**	2.11	2.13
	B	8.32	**3.16**	4.93	4.00
$img2$	R	49.75	9.78	**8.03**	8.24
	G	11.86	6.46	3.55	**3.30**
	B	52.05	11.15	13.31	**9.91**
$img3$	R	11.92	**3.11**	3.49	4.07
	G	3.52	**1.77**	2.31	2.54
	B	13.08	**4.06**	6.63	5.55
$img4$	R	125.71	20.49	18.34	**16.77**
	G	34.56	12.18	7.55	**6.38**
	B	128.39	19.81	23.10	**14.48**
$img5$	R	16.12	**3.31**	3.35	3.64
	G	4.87	2.07	1.89	**1.83**
	B	16.29	4.03	5.48	**3.98**
$img6$	R	35.07	8.31	**6.89**	8.46
	G	8.53	5.15	3.87	**3.34**
	B	35.14	7.20	7.67	**6.93**
$img7$	R	23.21	4.96	4.59	**4.41**
	G	5.50	3.21	**1.77**	1.78
	B	23.48	5.47	6.29	**4.97**
$img8$	R	43.26	6.80	6.72	**5.95**
	G	11.41	4.53	2.77	**2.55**
	B	45.16	7.99	7.77	**6.90**
$img9$	R	19.85	6.88	10.63	**5.48**
	G	5.63	5.12	**2.26**	2.87
	B	23.38	34.75	**7.76**	7.77
$img10$	R	39.72	8.99	**7.48**	8.15
	G	9.98	6.49	3.37	**3.34**
	B	42.19	9.96	**9.40**	9.47
$img11$	R	24.38	7.75	**7.10**	8.75
	G	7.57	4.72	4.76	**4.58**
	B	28.67	**8.80**	11.00	9.54
$img12$	R	58.38	20.71	16.35	**14.71**
	G	15.65	13.99	8.30	**7.65**
	B	74.19	25.69	24.42	**20.63**

TABLE III
S-CIELab Measure (ΔE^*_{ab}) Comparison of Different Demosaicing Methods

Image	Scheme [7]	Scheme [8]	Scheme [17]	This work
$img1$	0.5360	**0.4899**	0.5720	0.5395
$img2$	0.9624	0.8907	1.0154	**0.8833**
$img3$	0.6039	**0.5427**	0.6999	0.6818
$img4$	1.4441	**1.3147**	1.4266	1.3267
$img5$	0.5866	**0.5447**	0.6552	0.5834
$img6$	0.8125	**0.7418**	0.8000	0.7613
$img7$	0.6993	0.6592	0.7985	**0.6520**
$img8$	0.7941	0.7536	0.7985	**0.7383**
$img9$	0.6687	0.6370	0.6577	**0.6312**
$img10$	0.9609	**0.8643**	0.8901	0.8655
$img11$	0.9401	**0.8647**	0.9734	0.9536
$img12$	1.1438	**1.0108**	1.0998	1.0961

Two objective quality measures are used to evaluate the performance of demosaicing algorithms: MSE metric and S-CIELab metric ΔE^*_{ab} [22], [8]. For color images, human vision system-based S-CIELab metric is more appropriate than MSE metric. However, since MSE metric has been widely used in the literature of CFA demosaicing, we choose to report both. Additionally, we include subjective quality comparison to support the effectiveness of our new demosaicing technique. The readers are suggested to visit the above website to evaluate the visual quality of reconstructed images by various demosaicing techniques on a monitor.

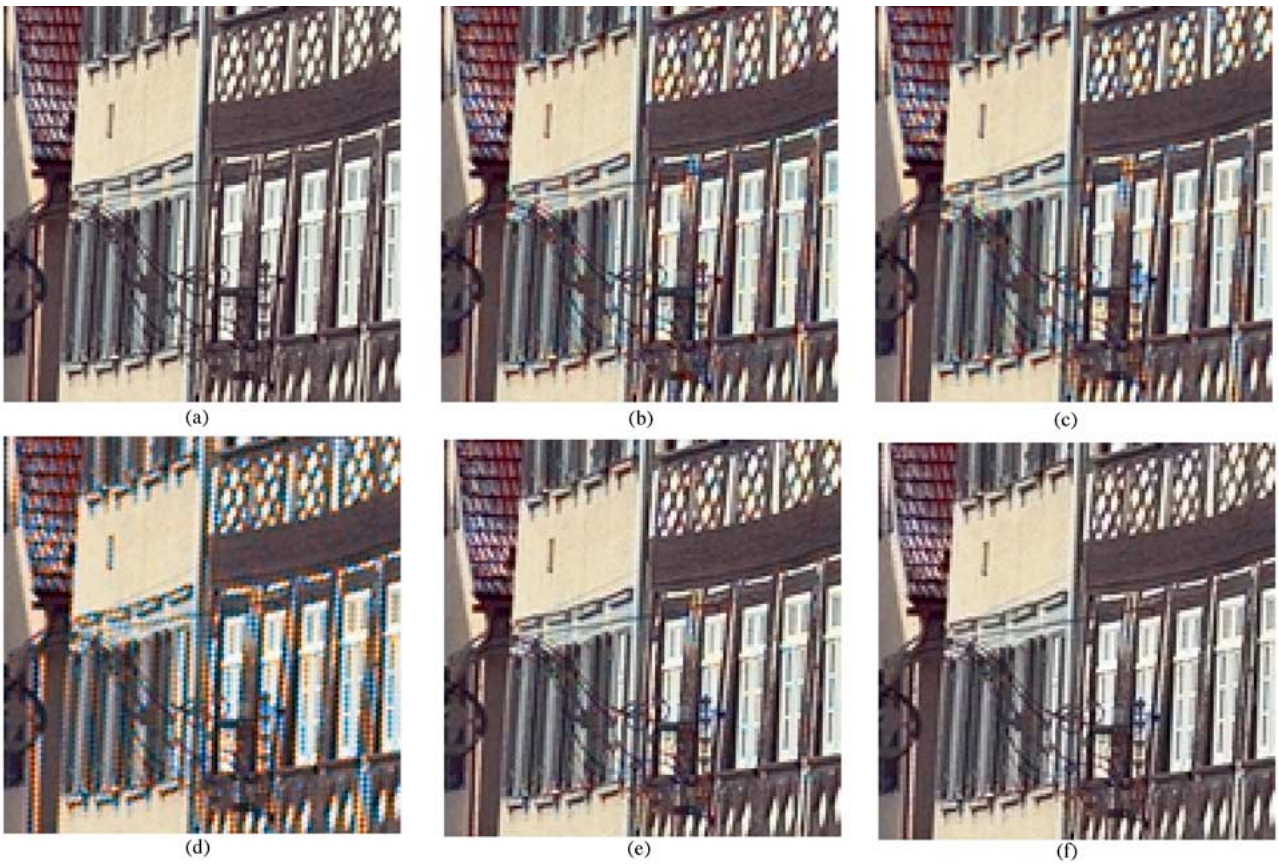

Fig. 11. (a) Portion of original *img*4. (b) Reconstructed image by [17]. (c) Reconstructed image by [8]. (d) Reconstructed image by [7]. (e) Our scheme (universal threshold $\delta_l = 4$). (f) Our scheme (spatially adaptive threshold $\delta_l = 4$ and $\delta_h = 0.05$).

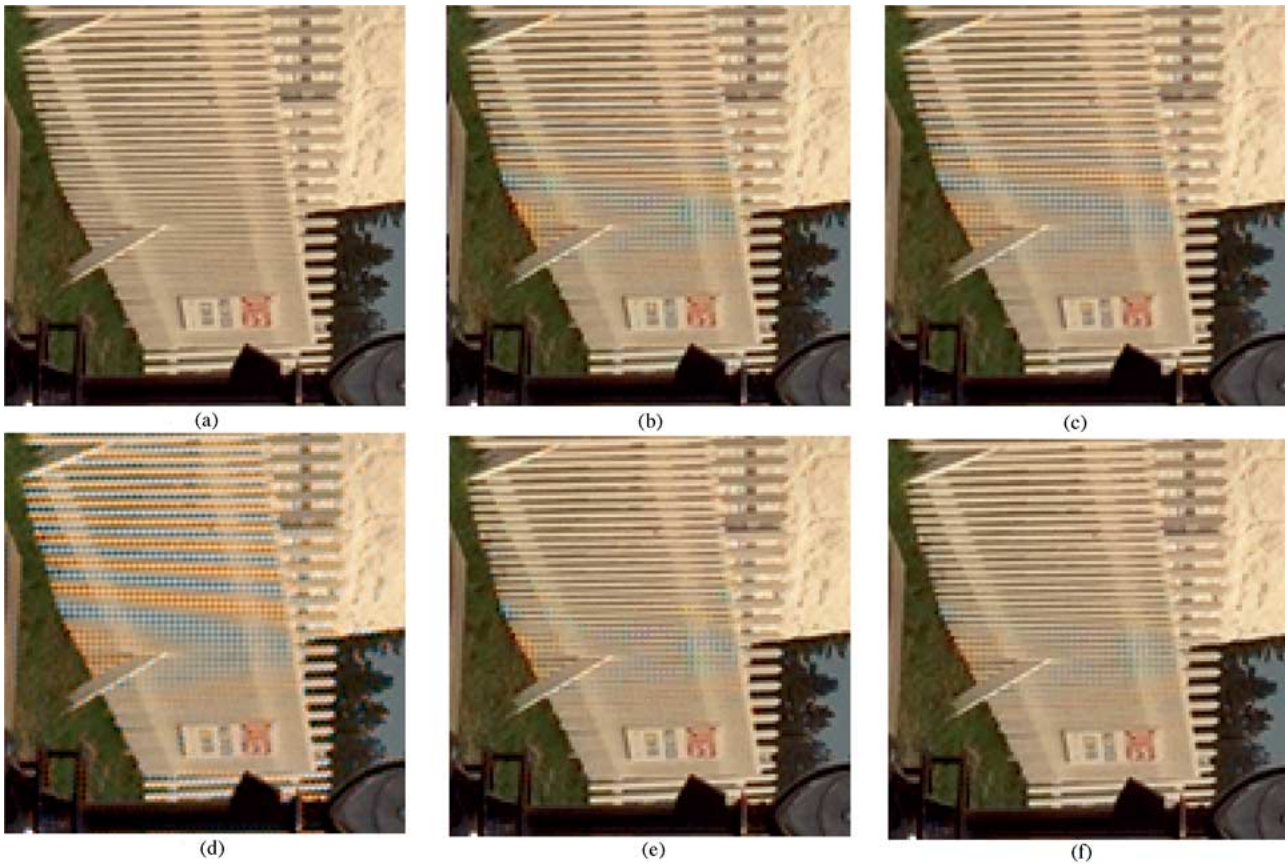

Fig. 12. (a) Portion of original *img*8. (b) Reconstructed image by [17]. (c) Reconstructed image by [8]. (d) Reconstructed image by [7]. (e) Our scheme (universal threshold $\delta_l = 4$). (f) Our scheme (spatially adaptive threshold $\delta_l = 4$ and $\delta_h = 0.05$).

A. Initial Condition and Convergence Property

To illustrate the sensitivity of iterative demosaicing to the initial condition, three different intraplane interpolation techniques have been tested for the green pixels: bilinear interpolation, edge-sensitive interpolation [4], and edge-directed interpolation [14]. Fig. 9 compares the evolution of MSE under the first two initial conditions: bilinear and edge sensitive (the third [14] achieves very similar result to [4], but it is much more computationally expensive). It can be observed that edge-sensitive interpolation does lead to smaller MSE than bilinear at the price of slightly increased complexity. Therefore, we suggest the use of edge-sensitive interpolation at the initialization.

To see the convergence property of the proposed iterative demosaicing, we show in Fig. 10 the evolution of MSE terms E_r, E_g, E_b as the function of iteration number n (we plot $\log(E)$ instead of E in the figure to facilitate visual inspection). It can be verified that all those terms monotonically decrease as the iteration number increases. Such observation validates the use of a threshold to terminate the iteration. We note that comparing Figs. 9 and 10 clearly shows that the limiting solution does not render the smallest MSE. It is the intelligent choice of stopping criterion that achieves nearly optimal MSE result and avoids the potential zipper artifacts.

B. Performance Comparison With the Benchmarks

Two recent image demosaicing techniques are used as the benchmarks: [8] and [17]. Since the work in [7] corresponds to our scheme with exact one iteration and a simplified starting point (bilinear interpolation), we include [7] into the comparison, as well. Therefore, the gain over [7] is precisely due to the idea of iterative demosaicing and improved initial condition. The work in [17] represents another class of iterative demosaicing using convex-set projection theory. Our scheme and [17] share the same initial condition (edge-sensitive interpolation for green pixels) but employ different iterative updating strategies. The most recent work [8] emphasizes the suppression of demosaicing artifacts by post processing in a noniterative framework.

Tables II and III compare the MSE and S-CIELAB performance among different schemes respectively. We use bold-type font to highlight the smallest MSE value across each row. It can be observed that our algorithm achieves the best MSE performance in most situations. On S-CIELAB performance, ours appears to be comparable with scheme [8] (the post-processing step in [8] contributes to its excellent ΔE^*_{ab} performance). We also note that in this experiment, a universal threshold value $\delta_l = \delta_h = 4$ is chosen for all pixels in our scheme. Such a strategy is not the best for the subjective quality of reconstructed images but has been found nearly optimal on objective performance.

To appreciate the effectiveness of the proposed spatially adaptive stopping criterion on suppressing artifacts (i.e., from a universal threshold to two threshold values $\delta_l = 4, \delta_h = 0.05$), we use *img*4 and *img*8—two images with relatively large amount of aliasing, which has caused tremendous difficulty to CFA demosaicing. Figs. 11 and 12 compare the demosaiced images by the benchmark schemes and our iterative demosaicing with and without spatially adaptive stopping criterion. We observe that color misregistration artifacts have been successfully suppressed by the two-class stopping criterion, especially for *img*8. By contrast, demosaiced images by the existing techniques [17], [8] still suffer from noticeable artifacts due to color misregistration.

C. Complexity Cost and Memory Requirement

It is straightforward to estimate the computational cost of the proposed demosaicing algorithm. Since we use the same edge-sensitive interpolation as [17], the cost at the starting point is the same. At each iteration, it takes $2MN$ additions and $1.25MN$ multiplications to update *R*, *G*, and *B* channels according to (9)–(16). The number of iterations is typically in the range of 3–5 for the test images, which amounts to a total of 10–15 MN arithmetic operations. Iterative method in [17] requires about 480 MN additions and multiplications due to wavelet filtering involved at each iteration. Iterative method in [16] is also computationally expensive due to the calculation of the color ratio and edge indicator function (both require nonlinear operations) at each iteration.

We have also compared the memory requirement for different iterative demosaicing schemes. Since our scheme is directly built upon edge-sensitive interpolation [4] and color-difference interpolation [7], the memory requirement is about the same (note that color plane updating can be implemented by in-place operations). According to [7], when edge-sensitive or color-difference interpolation is implemented on a DSP, it requires triple the number of memory access compared with bilinear interpolation. Such requirement is approximately 25% lower than that of alternating projection scheme [17] with one-level wavelet transform. As the level number increases, [17] requires more memory space to store high-band wavelet coefficients.

The new demosaicing algorithm has been implemented using both MATLAB and C. The running time is reported with respect to a Pentium-IV Dell Laptop (2.4 GHz). We have found that the actual running time of iterative demosaicing is typically 5–10 s, including the initialization (edge-sensitive interpolation). By contrast, it takes the alternating projection scheme [17] 10–15 s. Despite being a noniterative scheme, the software implementation of [8] appears to take much longer time (several minutes). The C-code implementation of our iterative demosaicing algorithm runs even faster and the overall processing time is less than a second for a 512×768 image.

VI. Concluding Remarks

In this paper, we present a new, fast, and efficient demosaicing technique. It is argued that the linear interpolation model works more effectively on color difference than color ratio signals. Our approach successively refines the estimate of missing data by enforcing the color difference rule at each iteration. We also investigate the cause of color misregistration and zipper artifacts in iterative demosaicing and propose an effective spatially adaptive stopping criterion to suppress both types of artifacts. Extensive experiment results are used to demonstrate the effectiveness of our new demosaicing technique. The proposed scheme is capable of achieving at least comparable and often better performance than existing iterative demosaicing techniques at the cost of lower computational complexity.

APPENDIX

In this appendix, we show that the constraint set defined by the linear interpolative model in the color difference domain is convex. Such a set can be written by $C = \{D_{a,b}(x) | D_{a,b}(x) = 1/|\mathcal{N}_x| \sum_{y \in \mathcal{N}_x} D_{a,b}(y)\}$, where $\mathcal{N}_x$ denotes the local neighborhood of x.

Proof

Let us take two points from set C

$$D^1_{a,b}(x) = \frac{1}{|\mathcal{N}_x|} \sum_{y \in \mathcal{N}_x} D^1_{a,b}(y), D^2_{a,b}(x) = \frac{1}{|\mathcal{N}_x|} \sum_{y \in \mathcal{N}_x} D^2_{a,b}(y)$$

then, for an arbitrary point between them, i.e.

$$\begin{aligned} D^3_{a,b}(x) &= \lambda D^1_{a,b}(x) + (1-\lambda) D^2_{a,b}(x) \\ &= \lambda \left(\frac{1}{|\mathcal{N}_x|} \sum_{y \in \mathcal{N}_x} D^1_{a,b}(y) \right) + (1-\lambda) \left(\frac{1}{|\mathcal{N}_x|} \sum_{y \in \mathcal{N}_x} D^2_{a,b}(y) \right) \\ &= \frac{1}{|\mathcal{N}_x|} \sum_{y \in \mathcal{N}_x} \left(\lambda D^1_{a,b}(y) + (1-\lambda) D^2_{a,b}(y) \right) \\ &= \frac{1}{|\mathcal{N}_x|} \sum_{y \in \mathcal{N}_x} D^3_{a,b}(y). \end{aligned}$$

Therefore, we conclude that C is a convex set.

ACKNOWLEDGMENT

The author would like to thank Dr. B. K. Gunturk at Louisiana State University and Dr. Y.-P. Tan at Nanyang Technological University for providing their demosaicing software as the benchmark.

REFERENCES

[1] B. E. Bayer, "Color imaging array," U.S. Patent 3 971 065, Jul. 1976.
[2] D. R. Cok, "Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal," U.S. Patent 4 642 678, Feb. 1987.
[3] J. E. Adams and J. F. Hamilton Jr., "Adaptive color plane interpolation in single color electronic camera," U.S. Patent 5 506 619, Apr. 1996.
[4] J. F. Hamilton Jr. and J. E. Adams, "Adaptive color plane interpolation in single color electronic camera," U.S. Patent 5 629 734, May 1997.
[5] A. Jain, *Fundamentals of Digital Image Processing*. Upper Saddle River, NJ: Prentice-Hall, 1989.
[6] J. E. Adams Jr., "Design of practical color filter array interpolation algorithms for digital cameras," *Proc. SPIE*, vol. 3028, pp. 117–125, Feb. 1997.
[7] S.-C. Pei and I.-K. Tam, "Effective color interpolation in CCD color filter arrays using signal correlation," *IEEE Trans. Circuits Systems Video Technol.*, vol. 13, no. 6, pp. 503–513, Jun. 2003.
[8] W. Lu and Y.-p. Tan, "Color filter array demosaicing: New method and performance measures," *IEEE Trans. Image Process.*, vol. 12, no. 10, pp. 1194–1210, Oct. 2003.
[9] J. W. Glotzbach *et al.*, "A method of color filter array interpolation with alias cancellation properties," in *Proc. Int. Conf. Image Process.*, vol. 1, 2001, pp. 141–144.
[10] E. Chang *et al.*, "Color filter array recovery using a threshold-based variable number of gradients," *Proc. SPIE*, vol. 3650, pp. 36–43, 1999.
[11] X. Wu, W. K. Choi, and P. Bao, "Color restoration from digital camera data by pattern matching," *Proc. SPIE*, vol. 3018, pp. 12–17, 1997.
[12] T. Kuno and H. Sugiura, "New interpolation method using discriminated color correlation for digital still camera," *IEEE Trans. Consum. Electron.*, vol. 45, no. 1, pp. 259–267, Feb. 1999.
[13] D. Taubman, "Generalized Wiener reconstruction of images from color sensor data using a scale invariant prior," in *Proc. Int. Conf. Image Process.*, vol. 3, 2000, pp. 801–804.
[14] X. Li and M. Orchard, "New edge directed interpolation," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, Oct. 2001.
[15] H. J. Trussell and R. E. Hartwig, "Mathematics for demosaicing," *IEEE Trans. Image Process.*, vol. 11, no. 3, pp. 485–492, Mar. 2002.
[16] R. Kimmel, "Demosaicing: Image reconstruction from CCD samples," *IEEE Trans. Image Process.*, vol. 8, no. 9, pp. 1221–1228, Sep. 1999.
[17] 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.
[18] S. Carrato, G. Ramponi, and S. Marsi, "A simple edge-sensitive image interpolation filter," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, 1996, pp. 711–714.
[19] J. Allebach and P. W. Wong, "Edge-directed interpolation," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, 1996, pp. 707–710.
[20] A. Tekalp, *Digital Video Processing*. Upper Saddle River, NJ: Prentice-Hall, 1995.
[21] N. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Upper Saddle River, NJ: Prentice-Hall, 1984.
[22] M. D. Fairchild, *Color Appearance Models*. Reading, MA: Addison-Wesley, 1997.

Xin Li (S'97–M'00) received the B.S. degree (with highest honors) in electronic engineering and information science from the University of Science and Technology of China, Hefei, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, in 1996 and 2000, respectively.

He was a member of the Technical Staff with Sharp Laboratories of America, Camas, WA, from August 2000 to December 2002. Since January 2003, he has been a faculty member with the Lane Department of Computer Science and Electrical Engineering, University of West Virginia, Morgantown. His research interests include image/video coding and processing.

Dr. Li received the Best Student Paper Award at the Conference of Visual Communications and Image Processing, San Jose, CA, in January 2001.