Available online at www.sciencedirect.com
ScienceDirect
Image and Vision Computing 25 (2007) 1495–1508
www.elsevier.com/locate/imavis

A new edge-adaptive demosaicing algorithm for color filter arrays

Chi-Yi Tsai, Kai-Tai Song *

Department of Electrical and Control Engineering, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 300, Taiwan

Received 16 December 2004; received in revised form 14 September 2006; accepted 19 December 2006

Abstract

A novel edge-adaptive demosaicing algorithm (EADA) is proposed in this paper to effectively reduce color artifacts in demosaiced images from a color filter array (CFA). The proposed algorithm aims to reduce the aliasing error in red and blue channels by exploiting high-frequency information of the green channel. To achieve this, color-difference based edge-adaptive filtering and post-processing schemes are designed to reproduce the color values by exploiting the green channel information. For green channel interpolation, any of the existing image interpolation methods can be used and combined with the proposed algorithm. Moreover, a new adaptive interpolation method is presented for reconstructing the green channel from CFA samples. We have compared this technique with two recently proposed demosaicing techniques: Gunturk's and Lu's methods. The experimental results show that EADA outperforms both of them in both PSNR values and CIELAB ΔE^*_{ab} measures.

Keywords: Image representation; Color reproduction; CFA demosaicing; Adaptive filtering; Color artifacts

1. Introduction

Digital color images from single-chip digital still cameras are obtained by interpolating the output from a color filter array (CFA). The CFA consists of a set of spectrally selective filters that are arranged in an interleaved pattern so that each sensor pixel samples one of three primary color components. These sparsely sampled color values are termed mosaiced images. To render a full-color image from a mosaiced image, an image reconstruction process, commonly known as CFA interpolation or CFA demosaicing, is required to estimate for each pixel its two missing color values. The simplest CFA demosaicing methodologies apply well-known interpolation techniques to each color channel such as nearest-neighbor replication, bilinear interpolation, and cubic spline interpolation. However, these single-channel algorithms usually introduce severe color artifacts and blurring around sharp edges [1]. These drawbacks indicate the need for more specialized algorithms for advanced demosaicing performance. We discuss these demosaicing methods in three groups.

The first class of demosaicing methods is based on high cross-correlation between color channels [1–3]. In [2], smooth hue transition algorithms are presented based on inter-channel correlation, which assumes that hue does not change abruptly between neighboring pixel locations. This class of algorithms exploits color ratios between red and green, blue and green to interpolate the missing color values. In addition to interpolating the color ratios, they also use inter-channel color differences (red minus green and blue minus green) [3]. Although these algorithms normally give better performance than single-channel algorithms, they cannot produce satisfactory demosaicing results around sharp edges, where the assumption of inter-channel correlation does not hold.

To reduce the undesirable demosaicing artifacts on sharp edges, the second class of demosaicing algorithms employs an edge-directed interpolation approach to perform the interpolation along image edges and prevent the interpolation across edges [4–7]. These methods first analyze the

* Corresponding author. Tel.: +886 3 5731865; fax: +886 3 5715998.
E-mail addresses: chiyi.ece91g@nctu.edu.tw (C.-Y. Tsai), ktsong@mail.nctu.edu.tw (K.-T. Song).

0262-8856/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2006.12.018

spatial correlation of a local image neighborhood and then select a suitable interpolation direction together with its neighboring pixels to estimate the missing color values. The third class of demosaicing methods uses both the above demosaicing approaches [8–15]. Kimmel proposes a hybrid demosaicing method [9], which estimates each missing color value by combining the edge-directed interpolation with smooth hue transition algorithm in an iterative process. Li and Orchard [10] proposed an edge-directed interpolation scheme to interpolate inter-channel color differences by exploiting the geometric duality between low-resolution covariance of CFA samples and high-resolution covariance of demosaiced images. In [11], a nonlinear demosaicing scheme based on optimal recovery interpolation of gray scale images was proposed. The optimal recovery interpolation scheme was presented to interpolate the green plane and the inter-channel color differences. Another effective method that exploits inter-channel correlation was proposed by Gunturk et al. [13]. They utilized an edge-directed interpolation scheme to estimate the missing color values in the green channel and an alternating projection scheme to estimate the missing color values in red and blue channels based on high inter-channel correlation. In a recent effort, Lu and Tan [14] presented an improved hybrid CFA demosaicing method that consists of two successive steps: an interpolation step to render full-color images and a post-processing step to suppress visible demosaicing artifacts.

The second and third class algorithms generally produce high quality visual results, especially in reconstructing sharp edges of the demosaiced image. However, in regions of fine details, where edges tend to be short and in different directions, these algorithms introduce undesirable errors and generate color artifacts. Color artifacts are caused primarily by aliasing error in high-frequency regions such as edges or fine textures. In demosaicing, color artifacts existing around edges and fine textures of the demosaiced image are a factor limiting performance. Existing algorithms are unable to resolve color artifacts in these regions effectively to obtain demosaiced results with high visual quality.

To effectively reduce color artifacts in demosaiced images, we here propose a novel edge-adaptive CFA demosaicing algorithm. The proposed demosaicing algorithm consists of color-difference based edge-adaptive low-pass filtering and post-processing schemes to reproduce color values by exploiting the green channel information for making high-frequency components of red and blue channels similar to the green channel. The red and blue channels are first reconstructed using bilinear interpolation and then edge-adaptive filtered in color-difference space. Subsequently, the visible color artifacts in high-frequency regions of the full-color demosaiced image can be reduced effectively in the post-processing step. Another advantage of the proposed algorithm is that any existing image interpolation methods can be combined with the proposed algorithm to reconstruct the green channel. Moreover, we also present a new adaptive interpolation method for reconstructing the green channel from CFA samples.

To evaluate the performance of the proposed demosaicing method, we adopted the PSNR and CIELAB ΔE^*_{ab} [16] to measure the fidelity of demosaiced images. Experimental results reveal that the proposed method performs satisfactorily on well-defined edges and effectively reduces visible color artifacts in fine details of the demosaiced images. We have compared this algorithm with two well-known demosaicing techniques [13,14]. The experimental results show that our method outperforms both of them, both in PSNR values and CIELAB ΔE^*_{ab} measures, and also gives superior demosaiced fidelity in visual comparison.

The rest of this paper is organized as follows: In Section 2, we will describe the motivation to study the color-difference based demosaicing algorithm. Section 3 presents the proposed color-difference based demosaicing algorithm. We then present a new adaptive interpolation method for reconstructing a green channel from CFA samples in Section 4. Section 5 presents the experimental results and compares the demosaicing results of the proposed method with other existent methods. Section 6 summarizes the contributions of this work.

2. Color-difference approach to demosaicing

Fig. 1 illustrates the most used CFA pattern, the Bayer pattern [17], where R, G and B denote, respectively, pixels having only red, green and blue color values. We limit our discussion in this paper to the Bayer pattern because it is so popular. In a Bayer pattern, green samples are obtained on a quincunx, while red and blue samples are obtained in rectangular lattices. The density of red and blue samples is one-half that of the green ones, and the aliasing error of high-frequency components in green channel is likely to be less than that in red and blue channels. Thus, a common problem in demosaicing is that the visible color artifacts in high-frequency regions are caused primarily by aliasing in the red and blue channels. Fortunately, there is usually high inter-channel correlation in high-frequency regions among red, green, and blue channels for natural color images [13]. This implies that the red, green, and blue

G	R	G	R	G
B	G	B	G	B
G	R	G	R	G
B	G	B	G	B

Fig. 1. Bayer color filter array pattern.

channels are quite similar at fine texture and edge locations with all three colors. Therefore, a valid assumption can be made that object boundaries are the same in all three color channels. In other words, the high-frequency regions are similar in all three channels and close to the high-frequency regions of the green channel. In order to validate this assumption, we utilize twenty-four natural images from the Kodak PhotoCD (see Fig. 2), which have been used as test images for several demosaicing studies [13–15].

Fig. 3 shows the flowchart for demonstrating the assumption of color-difference model mentioned above. The key concept is to replace the high-frequency components of red and blue planes by using those of green plane, and compare then the mean squared error (MSE) between the original and reconstructed color planes. A low-pass filter is utilized for red and blue planes and a high-pass filter for the green plane. We utilize 2-D ideal low-pass and high-pass filters in this procedure. Their transfer functions are given by [18]:

$$H_{\text{lowpass}}(u,v) = \begin{cases} 1 & \text{if } D(u,v) \leqslant D_0, \\ 0 & \text{if } D(u,v) > D_0, \end{cases} \quad \text{and}$$

$$H_{\text{highpass}}(u,v) = \begin{cases} 0 & \text{if } D(u,v) \leqslant D_0, \\ 1 & \text{if } D(u,v) > D_0, \end{cases}$$

where D_0 is a specified nonnegative quantity; and $D(u,v)$ is the distance from point (u,v) to the origin of the frequency plane. We set D_0 equal to 128 in this test. After filtering in each color plane, the new red and blue planes, $\overline{R}$ and $\overline{B}$, are reconstructed by adding the high-frequency components of the green plane to their low-frequency components respectively. Table 1 records the MSE comparison results of each step. The first and second columns show the MSE between original and low-pass filtered red (blue) planes R_{low} (B_{low}). The third and fourth columns show the MSE between original and reconstructed red (blue) planes $\overline{R}(\overline{B})$. From the

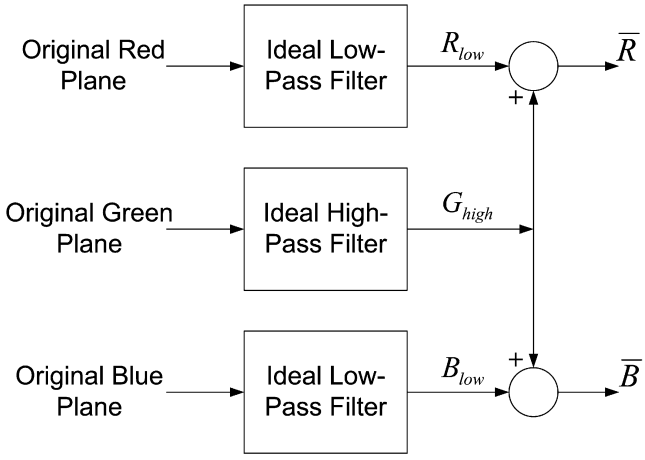

Fig. 3. Flowchart for demonstrating the assumption of color-difference model.

test results, it is clear that the MSE is reduced effectively by adding the high-frequency regions of the green plane G_{high} to the low-pass filtered red (blue) planes. This implies that the high-frequency regions of red and blue planes are similar and close to the high-frequency regions of the green plane. Thus, the assumption is validated. Based on this assumption, our motivation in this study is to reduce the color artifacts in high-frequency regions by adding the high-frequency information of green channel to other color channels. As described below, this can be achieved by utilizing the color-difference model.

Let $[R^d \quad G^d \quad B^d]$ denote three color planes of a demosaiced image. The Fourier spectrum of each color plane can be described as follows:

$$F[R^d] = F[R^d]_l + F[R^d]_h, \quad F[G^d] = F[G^d]_l + F[G^d]_h, \quad F[B^d] = F[B^d]_l + F[B^d]_h, \tag{1}$$

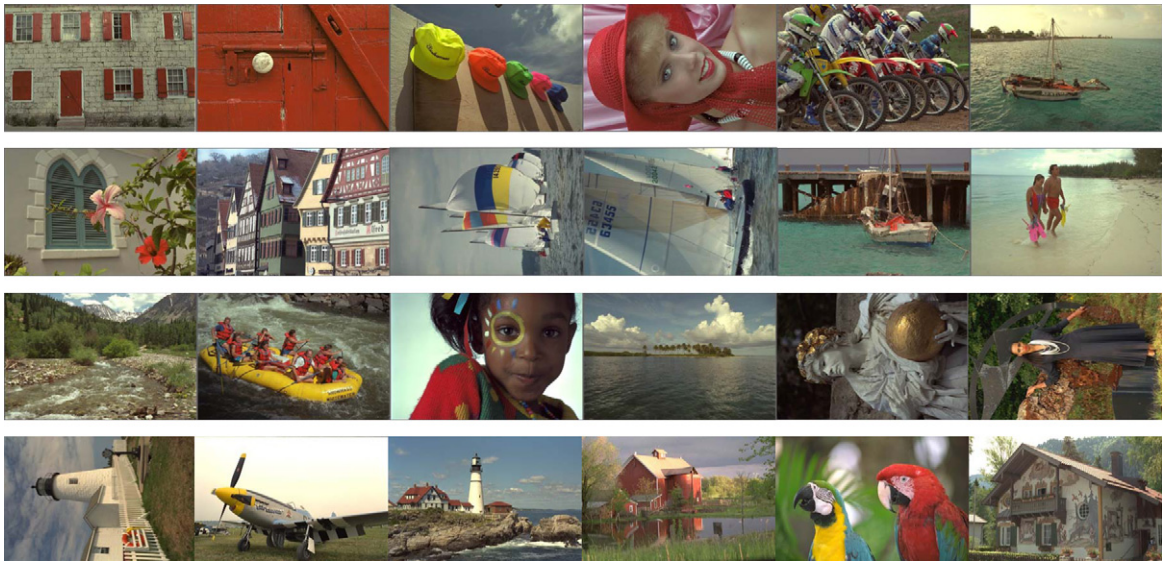

Fig. 2. Test images used in the experiment.

Table 1
Comparison of mean squared error at each step in Fig. 3

Image no.	MSE(R, R_{low})	MSE(B, B_{low})	MSE$(R, \overline{R})$	MSE$(B, \overline{B})$
1	260.2865	263.2530	4.2306	2.9049
2	73.0098	66.0994	9.8449	1.6476
3	59.1224	55.4568	3.1630	4.0863
4	75.0895	71.0032	8.0543	1.6137
5	307.8661	293.3846	6.3342	9.2040
6	178.8668	174.3140	2.1157	4.2843
7	76.3988	76.0140	2.8284	3.3626
8	539.2621	526.6953	8.2720	7.8362
9	81.2787	77.1119	2.6365	3.9927
10	86.7402	89.3761	2.8525	4.2737
11	149.5650	139.4246	4.1582	2.2200
12	70.4837	74.3636	3.1482	2.9343
13	415.0555	415.0555	3.3856	10.3724
14	168.1822	141.2964	9.1458	8.2062
15	112.1141	103.0222	12.2228	3.6253
16	82.9777	80.8168	1.3618	1.8066
17	81.5764	84.8154	2.2567	3.9789
18	187.1993	182.8144	6.1265	10.0651
19	179.4598	163.5454	2.7882	3.8415
20	119.7853	109.3889	2.4227	6.2912
21	155.7798	160.6295	2.6352	4.9741
22	124.3185	129.3350	7.7838	7.3834
23	57.6438	55.0500	4.2972	4.7799
24	232.1675	302.3846	10.3759	19.5268

where $F[\cdot]$ denotes the 2-D discrete Fourier transform; and the subscripts l and h stand for low-frequency and high-frequency components, respectively. The color-difference models of the demosaiced image are defined such that

$$R_g = R^d - G^d, \quad B_g = B^d - G^d. \tag{2}$$

Let $L\{\cdot\}$ denote a linear low-pass filtering process, and $\tilde{R}_g$ and $\tilde{B}_g$ denote the low-frequency regions of the color differences corresponding to R_g and B_g. Suppose that the high-frequency components of the color differences R_g and B_g can be removed by the low-pass filtering process, the Fourier spectrum of $\tilde{R}_g$ and $\tilde{B}_g$ can be described such that

$$\begin{aligned} F[\tilde{R}_g] &= L\{F[R_g]\} = F[R^d]_l - F[G^d]_l, \\ F[\tilde{B}_g] &= L\{F[B_g]\} = F[B^d]_l - F[G^d]_l. \end{aligned} \tag{3}$$

Subsequently, the new red and blue planes of the demosaiced image, $\overline{R}^d$ and $\overline{B}^d$, can be obtained by adding $\tilde{R}_g$ and $\tilde{B}_g$ with G^d respectively. Their Fourier spectra are given by

$$\begin{aligned} F[\overline{R}^d] &= F[\tilde{R}_g + G^d] = F[R^d]_l + F[G^d]_h, \\ F[\overline{B}^d] &= F[\tilde{B}_g + G^d] = F[B^d]_l + F[G^d]_h. \end{aligned} \tag{4}$$

It is clear from (4) that the high-frequency components of the new red and blue planes of the demosaiced image are replaced by the high-frequency components of the green plane. Because the aliasing error in the green plane is usually much smaller than those in red and blue planes, based on the assumption described above, the aliased errors in red and blue channels can be efficiently reduced by linear low-pass filtering in the color-difference spaces and adding the results with green channel to obtain the new ones. This observation leads to the development of an efficient demosaicing algorithm based on color-difference that can reduce the color artifacts in high-frequency regions such as edges or fine textures.

The color-ratio model has been another useful model for the development demosaicing algorithms [1,9]. The main difference between the color-difference model and the color-ratio model is that the latter assumes the ratios between the red and green values are constant within a given object, as are the ratios between the blue and green values. However, the color-ratio model usually fails to work around edge regions and results in some color artifacts because the constant-ratio assumption is not valid in these regions. But, again, the assumption used by color-difference model is that the high-frequency regions are similar in all three channels and close to the high-frequency regions of the green channel. If the high-frequency components of the green channel (such as edges and fine textures) can be recovered within small aliasing errors, then the results can be used to effectively reduce the aliasing errors in red and blue channels. In the following section, we describe the proposed edge-adaptive demosaicing algorithm (EADA) based on the color-difference model.

3. Proposed edge-adaptive demosaicing algorithm

Fig. 4 illustrates a simplified CFA demosaicing procedure for digital cameras. Given the CFA samples, an interpolation step is first performed to obtain the full color demosaiced image. Due to restrictions on computing power, the demosaiced images obtained from the interpolation step usually lack sharpness and contain false colors or color artifacts. Hence, a post-processing step is required to provide more visually pleasing demosaiced results. In the proposed EADA, the interpolation and post-processing steps are both developed based on the color-difference model.

To begin with derivation of the proposed demosaicing algorithm, we first assume that the green channel has been fully recovered by using an existing image interpolation method. The initial estimation of red and blue channels can be obtained, for instance, by utilizing some well-known method such as bilinear interpolation. Note that at this stage we do not process the original red (blue) values, but rather keep them the same as the original CFA-sampled color values. When an initial demosaiced image is obtained, we then utilize an adaptive low-pass filter to filter the color-difference value at the missing pixel locations in red (blue) channels.

3.1. Edge-adaptive low-pass filtering

Let $[R_i^d \quad G_i^d \quad B_i^d]$ denote three color planes of the initial demosaiced image. The color-difference planes are given by

$$R_g = R_i^d - G_i^d \quad \text{and} \quad B_g = B_i^d - G_i^d, \tag{5}$$

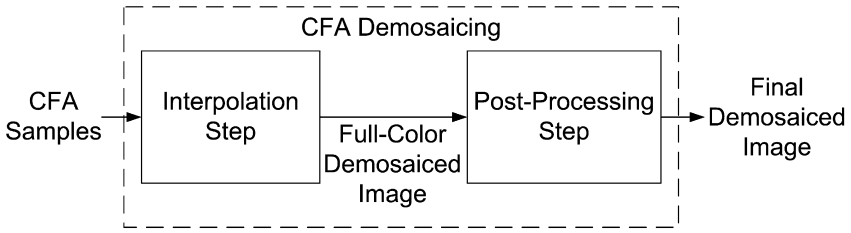

Fig. 4. Simplified CFA demosaicing procedure in digital cameras.

where R_i^d and B_i^d are, respectively, the initial estimated red and blue channels obtained from bilinear interpolation; and G_i^d is the initial estimated green channel obtained by any image interpolation method. The filtering procedure involves two sub-steps: first, edge-adaptive low-pass filtering of the red (blue) values over the original blue (red) pixels, as shown in Fig. 5(a); second, edge-adaptive low-pass filtering of the red (blue) values over the original green pixels, as shown in Fig. 5(b). Because the same procedure is used for both R_g and B_g color-difference planes, only the procedure of R_g will be described in the following presentation.

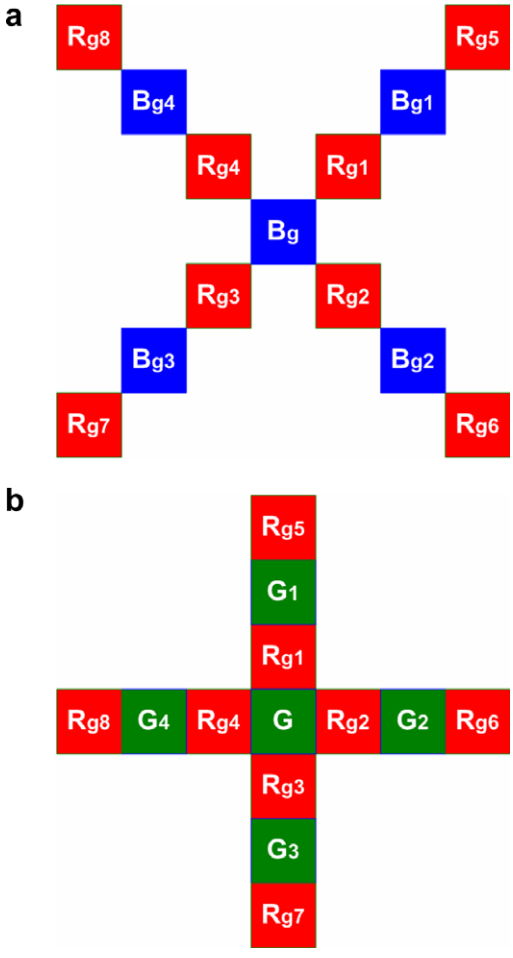

Fig. 5. (a) The red value on blue pixel and (b) the red value on green pixel of a central pixel to be estimated.

Referring to Figs. 5(a) and (b), the color-difference value R_g of red pixel at the center position is to be filtered adaptively by

$$R_g = \frac{e_{a1}\hat{R}_{g1} + e_{a2}\hat{R}_{g2} + e_{a3}\hat{R}_{g3} + e_{a4}\hat{R}_{g4}}{e_{a1} + e_{a2} + e_{a3} + e_{a4}}, \tag{6}$$

where $\hat{R}_{g1} \sim \hat{R}_{g4}$ are the color-difference adjusted values and $e_{a1} \sim e_{a4}$ are the edge indicators corresponding to each color-difference adjusted value. These edge indicators are defined as a decreasing function of the directional derivative of the center point and its neighboring points. In the edge-adaptive filtering stage, we propose to introduce the second-order directional derivatives of neighboring color-difference values for detecting edges more accurately. In the case of Fig. 5(a), the edge indicators are given by

$$e_{a1} = \frac{1}{1 + \left|\frac{R_{g1}-R_{g3}}{2\sqrt{2}}\right| + \left|\frac{R_{g5}-2R_{g1}+R_{g3}}{2\sqrt{2}}\right|}, \quad e_{a2} = \frac{1}{1 + \left|\frac{R_{g4}-R_{g2}}{2\sqrt{2}}\right| + \left|\frac{R_{g4}-2R_{g2}+R_{g6}}{2\sqrt{2}}\right|},$$
$$e_{a3} = \frac{1}{1 + \left|\frac{R_{g1}-R_{g3}}{2\sqrt{2}}\right| + \left|\frac{R_{g1}-2R_{g3}+R_{g7}}{2\sqrt{2}}\right|}, \quad e_{a4} = \frac{1}{1 + \left|\frac{R_{g4}-R_{g2}}{2\sqrt{2}}\right| + \left|\frac{R_{g8}-2R_{g4}+R_{g2}}{2\sqrt{2}}\right|}. \tag{7.a}$$

In the case of Fig. 5(b), they are given by

$$e_{a1} = \frac{1}{1 + \left|\frac{R_{g3}-R_{g1}}{2}\right| + \left|\frac{R_{g3}-2R_{g1}+R_{g5}}{2}\right|}, \quad e_{a2} = \frac{1}{1 + \left|\frac{R_{g2}-R_{g4}}{2}\right| + \left|\frac{R_{g6}-2R_{g2}+R_{g4}}{2}\right|},$$
$$e_{a3} = \frac{1}{1 + \left|\frac{R_{g3}-R_{g1}}{2}\right| + \left|\frac{R_{g7}-2R_{g3}+R_{g1}}{2}\right|}, \quad e_{a4} = \frac{1}{1 + \left|\frac{R_{g2}-R_{g4}}{2}\right| + \left|\frac{R_{g2}-2R_{g4}+R_{g8}}{2}\right|}. \tag{7.b}$$

The color-difference adjusted values $\hat{R}_{g1} \sim \hat{R}_{g4}$ are derived based on the assumption that the difference of neighboring color-difference values along an interpolation direction is constant. For example, to find the color-difference adjusted value at R_{g1} location, this assumption gives the following relationships for the neighboring color-difference values along the right-up direction

$$R_{g1} - R_{g3} = (R_{g1} - \hat{R}_g) + (\hat{R}_g - R_{g3}) \quad \text{and} \tag{8}$$
$$R_{g1} - \hat{R}_g = \hat{R}_g - R_{g3}, \tag{9}$$

where $\hat{R}_g$ denotes the missing color-difference value at B_g location. Combining (8) and (9), we have

$$R_{g1} - R_{g3} = 2(R_{g1} - \hat{R}_g).$$

This implies that $\hat{R}_g = R_{g1} + (R_{g3} - R_{g1})/2$. This value is denoted by $\hat{R}_{g1}$, which is used to estimate R_g in the right-up interpolation direction. In a similar manner, the

color-difference adjusted values along other interpolation directions are given by

$$\hat{R}_{g2} = R_{g2} + \frac{R_{g4} - R_{g2}}{2}, \quad \hat{R}_{g3} = R_{g3} + \frac{R_{g1} - R_{g3}}{2},$$
$$\hat{R}_{g4} = R_{g4} + \frac{R_{g2} - R_{g4}}{2}.$$

Finally, the full-red plane is obtained by recovering the spatial plane from color-difference plane such that

$$\overline{R}_i^d = R_g + G_i^d. \tag{10}$$

As the same procedure is utilized for recovering the blue plane, a full-color demosaiced image can be obtained.

3.2. Post-processing

The post-processing step aims to suppress visible color artifacts residing in the demosaiced image obtained from the edge-adaptive low-pass filtering step. To achieve this, a similar procedure is applied iteratively to smooth the color-difference values and make them become locally constant within an object. Let $[\overline{R}_i^d \quad G_i^d \quad \overline{B}_i^d]$ denote three color planes of the full-color demosaiced image obtained from the interpolation stage explained above. We first correct the green values to fit the color-difference values as smoothly as possible by using edge-adaptive filtering. The color-difference plane between green and red (blue) planes are given by

$$G_r = G_i^d - \overline{R}_i^d \quad \text{and} \quad G_b = G_i^d - \overline{B}_i^d. \tag{11}$$

Let us first illustrate how the color-difference value G_r in Fig. 6 is filtered by using the edge-adaptive filtering method as an example.

Referring to Fig. 6, the color-difference value G_r of the central pixel is to be filtered. Similar to the edge-adaptive interpolation, the color-difference value G_r is estimated by its neighboring color-difference values such that

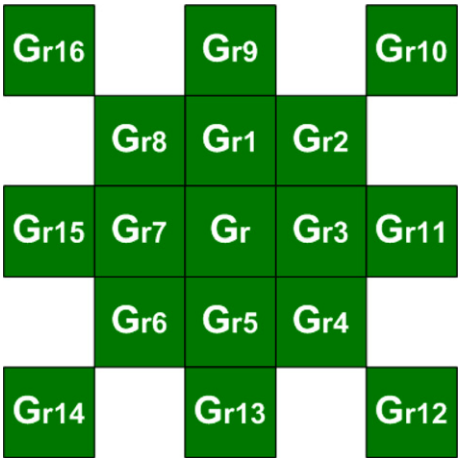

Fig. 6. Adaptive filtering of the red-green color difference plane.

$$G_r = \sum_{j=1}^{8} w_j G_{rj}, \quad w_j = \frac{e_{bj}}{\sum_{k=1}^{8} e_{bk}}, \tag{12}$$

where the edge indicators $e_{b1} \sim e_{b8}$ associated with the neighboring color-difference values $G_{r1} \sim G_{r8}$ are also defined as a decreasing function of the directional derivative of the center point and its neighboring points. However, at this stage, only the first-order directional derivatives of neighboring color-difference values are introduced. Experimentally, this can work better than the case of introducing the second-order directional derivatives. Thus, the edge indicators corresponding to each neighboring color difference value at this stage are given by

$$e_{b1} = \frac{1}{1 + \left|\frac{G_{r5} - G_{r1}}{2}\right| + \left|\frac{G_r - G_{r9}}{2}\right|}, \quad e_{b2} = \frac{1}{1 + \left|\frac{G_{r2} - G_{r6}}{2\sqrt{2}}\right| + \left|\frac{G_{r10} - G_r}{\sqrt{2}}\right|},$$
$$e_{b3} = \frac{1}{1 + \left|\frac{G_{r3} - G_{r7}}{2}\right| + \left|\frac{G_{r11} - G_r}{2}\right|}, \quad e_{b4} = \frac{1}{1 + \left|\frac{G_{r8} - G_{r4}}{2\sqrt{2}}\right| + \left|\frac{G_r - G_{r12}}{\sqrt{2}}\right|},$$
$$e_{b5} = \frac{1}{1 + \left|\frac{G_{r5} - G_{r1}}{2}\right| + \left|\frac{G_{r13} - G_r}{2}\right|}, \quad e_{b6} = \frac{1}{1 + \left|\frac{G_{r2} - G_{r6}}{2\sqrt{2}}\right| + \left|\frac{G_r - G_{r14}}{\sqrt{2}}\right|},$$
$$e_{b7} = \frac{1}{1 + \left|\frac{G_{r3} - G_{r7}}{2}\right| + \left|\frac{G_r - G_{r15}}{2}\right|}, \quad e_{b8} = \frac{1}{1 + \left|\frac{G_{r8} - G_{r4}}{2\sqrt{2}}\right| + \left|\frac{G_{r16} - G_r}{\sqrt{2}}\right|}. \tag{13}$$

Subsequently, the same procedure is applied to adaptively filter the color-difference plane between green and blue planes G_b. Then the post-processed green plane can be obtained such that

$$\overline{G}_p^d = \frac{(G_r + \overline{R}_i^d) + (G_b + \overline{B}_i^d)}{2}. \tag{14}$$

Since the same procedure is used to filter the other color-difference planes, $R_{gp} = \overline{R}_i^d - \overline{G}_p^d$ and $B_{gp} = \overline{B}_i^d - \overline{G}_p^d$, the post-processed red and blue planes can be obtained respectively as $\overline{R}_p^d = R_{gp} + \overline{G}_p^d$ and $\overline{B}_p^d = B_{gp} + \overline{G}_p^d$. This procedure is set to repeat three times, and the post-processed demosaiced images $[\overline{R}_p^d \quad \overline{G}_p^d \quad \overline{B}_p^d]$ will be completed after this stage.

Fig. 7 summarizes the proposed edge-adaptive demosaicing algorithm. For green plane interpolation, any of the existing image interpolation methods can be adopted, for instance edge-directed interpolation [10], adaptively quadratic image interpolation [19], etc. The initial full-color demosaiced image is first reconstructed in the interpolation step, then the color artifacts in the initial demosaiced result are reduced in the post-processing step. Note that the main difference between the proposed algorithm and Kimmel's method [9] is that Kimmel's method is based on the color-ratio model, whereas the proposed algorithm is based on the color-difference model. As a result, Kimmel's method usually induces color artifacts like overshot points around edge regions, but the proposed algorithm will not.

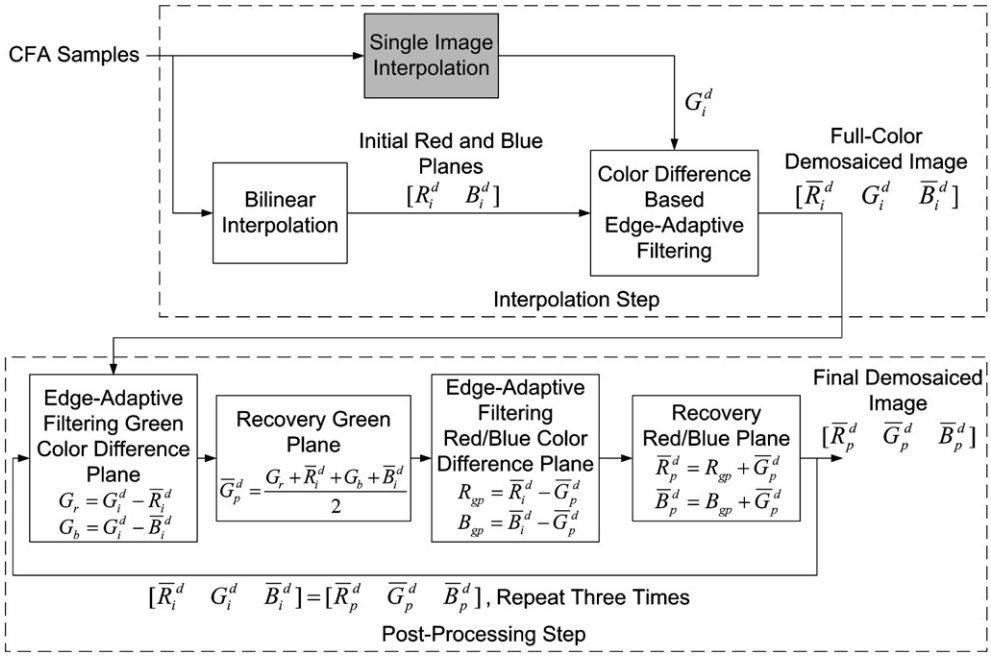

Fig. 7. Complete steps of the proposed edge-adaptive demosaicing algorithm.

4. Green channel adaptive interpolation

In this section, we present a novel adaptive interpolation method for green channel reconstruction from CFA samples. The green plane has the most spatial information of the image to be demosaiced and has great influence on the perceptual quality of the image. In order to reconstruct the demosaiced images with satisfactory quality, we propose a nonlinear procedure for choosing the direction of interpolation to reconstruct the green channel. Fig. 8 shows two cases of green samples in Bayer pattern, where the green value of central pixels are to be estimated from its four surrounding green pixels, $G_1 \sim G_4$. The central missing green value G can be estimated by the following expression:

$$G = \begin{cases} w_{s1}\hat{G}_1 + w_{s2}\hat{G}_2 + w_{s3}\hat{G}_3 + w_{s4}\hat{G}_4 & \text{if } E < T \\ \frac{w_{e1}\hat{G}_1 + w_{e3}\hat{G}_3}{w_{e1}+w_{e3}} & \text{if } E \geqslant T \text{ and } w_{e1} + w_{e3} > w_{e2} + w_{e4} \\ \frac{w_{e2}\hat{G}_2 + w_{e4}\hat{G}_4}{w_{e2}+w_{e4}} & \text{if } E \geqslant T \text{ and } w_{e2} + w_{e4} > w_{e1} + w_{e3} \\ w_{e1}\hat{G}_1 + w_{e2}\hat{G}_2 + w_{e3}\hat{G}_3 + w_{e4}\hat{G}_4 & \text{otherwise} \end{cases} \tag{15}$$

where $E = \max\{(|G_1 - G_2| + |G_3 - G_4|)/2, \quad (|G_1 - G_4| + |G_2 - G_3|)/2\}$; $\hat{G}_1 \sim \hat{G}_4$ are the color-adjusted green values; $w_{s1} \sim w_{s4}$ and $w_{e1} \sim w_{e4}$ are, respectively, the associated weights when $E < T$ and $E \geqslant T$; and T is a threshold value. In other words, if $E < T$, we regard the central pixel as being in a smooth region. Otherwise, we regard the central pixel as being in an edge region. In the smooth region, the weights associated with four surrounding color-adjusted

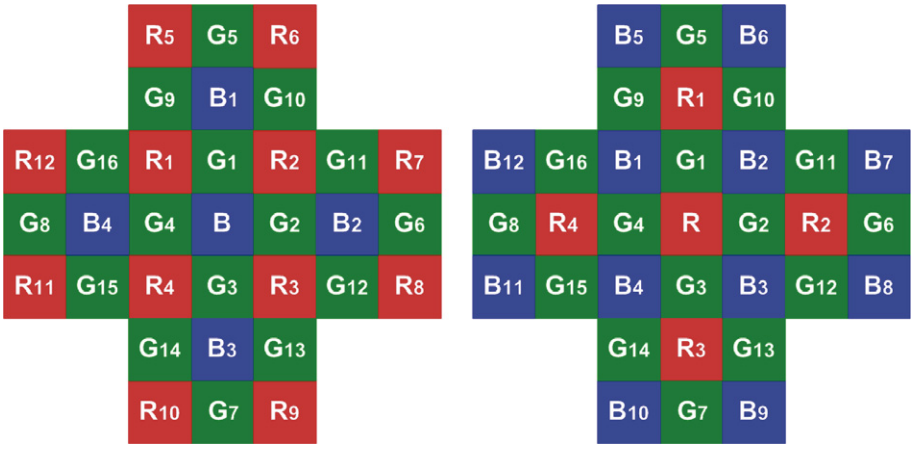

Fig. 8. Two cases of missing green value on the central pixel.

Table 2
PSNR (dB) and ΔE^*_{ab} measures for edge and smooth regions of demosaiced images

Step:	Interpolation step						Post-processing step					
Method:	Gunturk		Lu's interpolation.		EADA interpolation		Gunturk with Lu's post-processing		Lu's post-processing		EADA post-processing	
Region:	Smooth	Edge	Smooth	Edge	Smooth	Edge	Smooth	Edge	Smooth	Edge	Smooth	Edge
1	***34.8027***	***32.1314***	33.6657	30.2769	33.8758	29.6090	35.1747	32.5912	35.6250	32.5756	***36.3186***	***33.2922***
	1.3746	***1.8079***	1.4062	2.0304	***1.3657***	2.1142	1.3358	1.7421	1.2359	1.6944	***1.1649***	***1.5896***
2	36.1431	31.1341	37.3906	***31.6283***	***37.6166***	31.5044	36.0054	31.0292	37.3429	31.7327	***37.7334***	***32.2726***
	1.4912	2.4644	1.3306	2.3008	***1.2803***	***2.2252***	1.5067	2.4611	1.3316	2.3021	***1.2608***	***2.0880***
3	38.4495	31.2964	39.2719	***31.9197***	***39.8120***	31.7793	38.5750	31.3713	39.9506	32.5027	***40.3416***	***32.6216***
	0.8651	1.7615	0.7942	1.7531	***0.7562***	***1.6497***	0.8493	1.7614	0.7489	1.6302	***0.7333***	***1.5469***
4	36.3190	31.9110	***36.9232***	***32.0017***	36.9084	31.3759	36.2963	31.8991	***37.5835***	***32.8044***	37.2557	32.1889
	1.1891	1.9795	1.1411	***1.9963***	***1.1003***	2.0467	1.1444	1.9417	***1.0376***	***1.8063***	1.0647	1.8785
5	34.1087	***31.2459***	34.9989	30.7794	***35.1065***	30.0261	34.0323	31.2406	***35.7578***	***32.0784***	35.4706	31.9267
	1.9338	2.5203	1.6754	***2.5176***	***1.6324***	2.6263	1.9254	2.4951	***1.5695***	***2.2340***	1.6157	2.2673
6	***36.5753***	***31.7874***	35.6154	29.7878	36.1635	30.2979	36.8764	32.1401	36.9993	31.5921	***37.8157***	***33.2342***
	0.9202	***1.6638***	0.9298	1.8882	***0.8893***	1.7289	0.9035	1.6233	0.8484	1.6461	***0.8077***	***1.4133***
7	38.6294	33.7006	40.1308	***34.2764***	***40.2486***	34.0593	38.4476	33.4629	40.3646	34.6075	***40.4996***	***34.7691***
	0.9213	1.8230	0.7834	1.6691	***0.7645***	***1.5926***	0.9349	1.8801	0.7673	1.6307	***0.7589***	***1.5522***
8	33.3108	***29.5236***	33.1422	28.0243	***33.4184***	28.5194	33.5960	29.8621	34.4226	29.7022	***34.8608***	***31.1762***
	1.3547	2.1025	1.3325	2.2654	***1.2842***	***2.0965***	1.3229	2.0440	1.1988	1.9914	***1.1366***	***1.7281***
9	38.4874	32.8135	38.8031	32.4653	***39.2684***	***33.1266***	38.9150	32.9617	***39.9149***	33.3885	39.9019	***34.3989***
	0.7596	1.4954	0.7328	1.5502	***0.6897***	***1.3135***	0.7132	1.4655	***0.6565***	1.4168	0.6665	***1.2580***
10	38.5628	***33.1149***	38.8117	32.8390	***39.1464***	32.6105	38.9668	33.2002	***39.7257***	33.3782	39.6026	***33.7723***
	0.7898	1.3175	0.7636	1.3190	***0.7249***	***1.2743***	0.7439	1.2920	***0.6857***	1.2220	0.7081	***1.1959***
11	36.9033	***31.8417***	36.9056	30.6894	***37.1815***	30.3017	36.9826	32.0232	37.9296	32.3035	***38.1298***	***32.7767***
	1.2458	***1.9452***	1.1950	2.0457	***1.1602***	2.0448	1.1977	1.9036	***1.0793***	1.8103	1.0921	***1.6977***
12	39.1826	***34.3383***	39.3036	33.0570	***39.7133***	33.5898	39.3271	34.4150	40.1382	34.1477	***40.4659***	***35.3664***
	0.5996	***1.0088***	0.5785	1.1128	***0.5530***	1.0194	0.5932	1.0016	0.5448	1.0167	***0.5309***	***0.8983***
13	***31.8056***	***28.7975***	30.2278	26.7313	30.3566	25.8615	32.2870	29.3140	32.3168	29.2384	***32.7580***	***29.5080***
	2.1099	***2.8467***	2.2708	3.1780	2.2762	3.4638	2.0477	2.7319	1.9532	2.6386	***1.9177***	***2.6314***
14	32.9901	28.7035	34.8480	***30.2273***	***34.9636***	29.8372	32.8061	28.5379	***35.1331***	***30.2789***	34.6815	29.9897
	1.6537	2.4062	1.4625	2.2368	***1.4178***	***2.2265***	1.6427	2.3920	***1.3783***	2.1108	1.3790	***2.0683***
15	35.9837	***30.5924***	36.5217	29.7825	***36.7017***	28.9349	35.8843	30.5443	36.7190	***30.7231***	***36.9546***	30.1210
	1.2814	2.3972	1.1846	***2.3966***	***1.1428***	2.4931	1.2702	2.4032	1.1478	***2.2702***	***1.1172***	2.3154
16	***39.3335***	***34.5606***	38.3122	32.2189	38.9726	33.4249	39.5908	34.9486	39.5813	34.0333	***40.6099***	***36.3883***
	0.7976	***1.4893***	0.8331	1.7895	***0.7844***	1.5286	0.7854	1.4417	0.7671	1.5512	***0.7102***	***1.2344***
17	38.8503	***31.8483***	38.5223	31.1176	***38.8687***	30.5316	39.3416	31.9986	39.7215	32.2573	***39.7808***	***32.2743***
	1.3907	***1.7518***	1.4002	1.8633	***1.3061***	1.8591	1.2749	1.6875	***1.2235***	1.6532	1.2364	***1.6166***
18	34.5751	***30.0828***	34.4625	29.0260	***34.6087***	28.2103	34.7779	30.2372	***35.5586***	***30.4966***	35.3887	30.3397
	2.0733	***2.8434***	1.9601	2.8745	***1.9380***	3.0160	2.0240	2.8043	***1.7932***	***2.6044***	1.9123	2.7108
19	37.2828	***32.0914***	36.8853	31.4886	***37.2846***	31.6989	37.6850	32.3111	38.2983	32.6429	***38.3925***	***33.7268***
	1.0452	***1.9021***	1.0660	1.9979	***1.0163***	1.9443	0.9878	1.8332	0.9398	1.7792	***0.9352***	***1.6448***
20	37.5354	***31.5288***	38.0283	31.0972	***38.2942***	30.4797	37.7964	31.6786	38.6400	***32.3599***	***38.7041***	31.8354
	0.8303	***2.1026***	0.7749	2.1710	***0.7491***	2.2497	0.7990	2.0676	0.7253	***1.9373***	***0.7233***	2.0185
21	***36.5895***	***31.0595***	36.2703	29.5479	36.5689	29.0086	36.7403	31.4046	37.3029	31.5061	***37.6804***	***32.0090***
	0.9485	***2.1630***	0.9451	2.3824	***0.9203***	2.5049	0.9422	2.0970	0.8757	2.0141	***0.8460***	***1.9569***
22	35.0825	30.0799	35.8485	***30.6104***	***35.8768***	29.8668	35.0234	30.0016	***36.1690***	30.7396	35.9781	***30.8209***

	1.2433	2.0937	1.1343	***1.9987***	***1.1193***	2.1092	1.2728	2.1192	***1.1225***	***1.9268***	1.1467	1.9641
23	38.5853	31.4166	39.6931	32.1713	***40.2631***	***32.3015***	38.4481	31.3307	39.7167	32.5611	***40.3612***	***33.0353***
	0.8794	1.7241	0.8158	1.6618	***0.7731***	***1.5214***	0.8991	1.7492	0.8171	1.5970	***0.7687***	***1.4526***
24	33.2855	***26.8785***	***33.7499***	26.0904	33.6136	25.1634	33.6245	26.9839	34.1117	26.8372	***34.1219***	***26.9459***
	1.0799	***2.2544***	***0.9865***	2.2993	0.9954	2.4805	1.0719	2.2477	***0.9406***	***2.1398***	0.9844	2.1681
Avg.	36.3906	***31.3533***	36.5972	30.7439	***36.8680***	30.5050	36.5500	31.4787	37.4593	31.8537	***37.6587***	***32.2829***
	1.1991	***1.9943***	1.1457	2.0541	***1.1100***	2.0470	1.1745	1.9661	1.0579	1.8593	***1.0507***	***1.7873***

green values are denoted by $w_{s1} \sim w_{s4}$ and the central missing green value is then estimated by weighted sum of them. In the edge region, the weights associated with four surrounding color-adjusted green values are denoted by $w_{e1} \sim w_{e4}$ and the central missing green value is then carried out by selecting weighted sum in horizontal and vertical directions.

Based on (15), the color-adjusted green values and the corresponding weights need to be determined for estimating the missing green value G. For instance, in the case of interpolating the missing green value on blue pixel positions, the color-adjusted values in each interpolation direction are referred to [14] and given by

$$\hat{G}_1 = G_1 + \frac{B - B_1}{2}, \quad \hat{G}_2 = G_2 + \frac{B - B_2}{2}, \\ \hat{G}_3 = G_3 + \frac{B - B_3}{2}, \quad \hat{G}_4 = G_4 + \frac{B - B_4}{2}. \tag{16}$$

And the corresponding weights in smooth and edge regions are, respectively, given by

$$w_{sj} = \frac{e_{sj}}{\sum_{k=1}^{4} e_{sk}} \text{ and } w_{ej} = \frac{e_{ej}}{\sum_{k=1}^{4} e_{ek}}, \quad \text{for } j = 1 \sim 4, \tag{17}$$

where e_{sj} and e_{ej} are the edge indicators associated with four surrounding green pixels G_j. In smooth regions, a valid assumption is that the directional derivatives of each color channel are small. Thus, the edge indicates in smooth regions can be seen as a decreasing function dependent on the sum of local first-order directional derivative such that

$$e_{s1} = \frac{1}{1 + |G_1 - G_3| + |G_5 - G_1| + |B_1 - B| + \left|\frac{G_9 - G_4 + G_{10} - G_2}{2}\right| + \left|\frac{R_5 - R_1 + R_6 - R_2}{2}\right|} \\ e_{s2} = \frac{1}{1 + |G_2 - G_4| + |G_6 - G_2| + |B_2 - B| + \left|\frac{G_{11} - G_1 + G_{12} - G_3}{2}\right| + \left|\frac{R_7 - R_2 + R_8 - R_3}{2}\right|} \\ e_{s3} = \frac{1}{1 + |G_1 - G_3| + |G_3 - G_7| + |B - B_3| + \left|\frac{G_2 - G_{13} + G_4 - G_{14}}{2}\right| + \left|\frac{R_3 - R_9 + R_4 - R_{10}}{2}\right|} \\ e_{s4} = \frac{1}{1 + |G_2 - G_4| + |G_4 - G_8| + |B - B_4| + \left|\frac{G_1 - G_{16} + G_3 - G_{15}}{2}\right| + \left|\frac{R_4 - R_{11} + R_1 - R_{12}}{2}\right|}. \tag{18}$$

In edge regions, the assumption is that the directional derivatives of each color channel are consistent along the direction of edges. Thus, the edge indicates in edge regions can be seen as a decreasing function dependent on the consistence of local first-order directional derivative such that

$$e_{e1} = \frac{1}{1 + |G_1 - G_3| + |G_5 - 2G_1 + G_3| + |B_1 - B - G_1 + G_3| + |R_5 - R_1 - G_9 + G_4| + |R_6 - R_2 - G_{10} + G_2|} \\ e_{e2} = \frac{1}{1 + |G_2 - G_4| + |G_6 - 2G_2 + G_4| + |B_2 - B - G_2 + G_4| + |R_7 - R_2 - G_{11} + G_1| + |R_8 - R_3 - G_{12} + G_3|} \\ e_{e3} = \frac{1}{1 + |G_1 - G_3| + |G_1 - 2G_3 + G_7| + |G_1 - G_3 - B + B_3| + |G_2 - G_{13} - R_3 + R_9| + |G_4 - G_{14} - R_4 + R_{10}|} \\ e_{e4} = \frac{1}{1 + |G_2 - G_4| + |G_2 - 2G_4 + G_8| + |G_2 - G_4 - B + B_4| + |G_1 - G_{16} - R_1 + R_{12}| + |G_3 - G_{15} - R_4 + R_{11}|}. \tag{19}$$

Once the weights of each color-adjusted green value are obtained from (17), the missing green value G can be obtained by using (15). Finally, the full green channel can be obtained by adopting the same procedure as described above to interpolate the missing green value on red pixel posi-

tions. This method for interpolating green channel from CFA samples is combined with the proposed demosaicing algorithm.

5. Experimental results

Extensive experimental studies have been carried out to verify the effectiveness of the proposed demosaicing algorithm. In the experiments, twenty-four natural images from the Kodak PhotoCD, as shown in Fig. 2, are employed to demonstrate the demosaicing performance. The demosaiced results of the proposed method are compared with the recently published methods of Gunturk [13] and Lu [14]. For Gunturk's method, we make use of one-level (1-L) decomposition with eight projection iterations in the experiments. Because Gunturk's method consists of only the interpolation step, for fair comparison in each step, Lu's post-processing method was adopted as the post-processing step for Gunturk's method. As shown in Fig. 1, all test images are down-sampled to obtain the Bayer pattern and then reconstructed using the demosaicing methods under comparison in RGB color space.

To evaluate the quality of the demosaiced images, PSNR and CIELAB ΔE^*_{ab} are adopted as performance measure. The PSNR (in dB) is defined as

$$PSNR(\mathrm{dB}) = 10\log_{10}\left\{255^2\left(\frac{1}{MN}\sum_{1\leqslant y\leqslant M}\sum_{1\leqslant x\leqslant N}\|\overline{O}(x,y)-\overline{D}(x,y)\|^2\right)^{-1}\right\}, \tag{20}$$

where M, N are the total row and column number of the image; $\overline{O}(x,y)$ is the color vector at the $(x, y)^{\text{th}}$ position of the original color image; and $\overline{D}(x,y)$ is the corresponding color vector in the demosaiced color image. The CIELAB ΔE^*_{ab} is given by

$$\Delta E^*_{ab} = \frac{1}{MN}\sum_{1\leqslant y\leqslant M}\sum_{1\leqslant x\leqslant N}\|\overline{O}_{Lab}(x,y)-\overline{D}_{Lab}(x,y)\|, \tag{21}$$

where $\overline{O}_{Lab}(x,y)$ and $\overline{D}_{Lab}(x,y)$ are the color vector in CIELAB color space at the $(x, y)^{\text{th}}$ position of the original and

Fig. 9. Zoom-in demosaicing test results of test image No. 1: (a) original picture, (b) demosaiced result of Gunturk's method, (c) Lu's interpolation result, (d) the proposed EADA interpolation result, (e) Gunturk's method with Lu's post-processing result, (f) Lu's post-processing result, and (g) the proposed EADA post-processing result.

demosaiced color image, respectively. A ΔE^*_{ab} greater than 2.3 indicates the color difference is visible, while greater than 10 means the demosaiced image is very different from the original one [16]. For a demosaiced image, high fidelity implies high PSNR and small CIELAB ΔE^*_{ab} measures.

Furthermore, we compute the PSNR and ΔE^*_{ab} measures of smooth and edge regions separately for more detailed comparison with other algorithms. The method used to divide the original image into smooth and edge regions is the same as that discussed in Section 4. The threshold value T was set to 10 in the experiments. Table 2 presents the PSNR values and CIELAB ΔE^*_{ab} for the smooth and edge regions of the demosaiced results obtained by the proposed interpolation method together with those by other methods for comparison. In each step, the bold-italic font denotes the largest PSNR and smallest CIELAB ΔE^*_{ab} measures of smooth and edge regions across each row. It can be seen from the table that EADA provides improved demosaiced fidelity for smooth regions in most images compared with other methods in the interpolation step. However, in the post-processing step, the proposed post-processing scheme not only obtains the best demosaiced results in the smooth regions, but also provides superior improvement in the edge regions, which are crucial to the visual quality. These improvements can add-up the PSNR and reduce ΔE^*_{ab} of the edge regions of the interpolation result by about 1.78 dB and 0.26 units, respectively in average. In some cases, the gap can be as large as 3.89 dB (PSNR) and 0.83 ΔE^*_{ab} units, e.g., for image No. 13.

Fig. 9 explains why such a large performance gap exists between the proposed post-processing method and others. Fig. 9(a) shows a zoom-in of the test image No. 1. Figs. 9(b)–(d) are, respectively, the demosaiced results obtained from Gunturk's, Lu's and the proposed EADA interpolation step. The corresponding PSNR and ΔE^*_{ab} measures for smooth and edge regions are presented separately in Table 3. It can be seen that these demosaiced images have different types of color artifacts around the edge and smooth regions. In Figs. 9(b) and (c), the color artifacts are mainly distributed in the smooth regions (such as the blinds of the window) and appeared as pale false-colors with some zipper-effect. In contrast, the color artifacts in Fig. 9(d) are concentrated around the edge regions and appeared as deep false-colors with little zipper-effect. Therefore, as can be seen from Table 3, the smooth regions of Fig. 9(d) give better PSNR and ΔE^*_{ab} measures than those of Figs. 9(b) and (c) in the interpolation step.

Figs. 9(e)–(g) are the post-processing results corresponding to Figs. 9(b)–(d), respectively. Furthermore, Figs. 9(e–f) and (g) present, respectively, the demosaiced results obtained by using Lu's and the proposed post-processing step. It is clear that Figs. 9(e-f) still retain some color artifacts in smooth and edge regions, which is mainly because Lu's post-processing is only applied to artifact-prone regions (mainly the edge regions) [14]. This implies that Lu's post-processing method cannot effectively reduce the color artifacts in smooth regions, which usually contain some fine textures. Thus, color artifacts in the smooth

Table 3
PSNR (dB) and ΔE^*_{ab} measures for a zoom-in of edge and smooth regions of demosaiced images

Step:	Interpolation step						Post-processing step					
Method:	Gunturk		Lu's interpolation		EADA interpolation		Gunturk with Lu's post-processing		Lu's post-processing		EADA post-processing	
Region:	Smooth	Edge	Smooth	Edge	Smooth	Edge	Smooth	Edge	Smooth	Edge	Smooth	Edge
Fig. 9	31.3975	***30.1370***	30.5873	29.3493	***31.6826***	29.0357	31.8194	30.4666	31.6927	31.1705	***34.4615***	***31.9662***
(No.1)	1.9181	***2.4378***	1.9463	2.4438	***1.5840***	2.4576	1.8461	2.3722	1.7926	2.1450	***1.3745***	***1.9994***
Fig. 10	31.8618	***26.7096***	30.5177	25.1262	***33.6837***	26.1956	32.6248	27.3324	32.3797	27.2304	***35.7136***	***29.5209***
(No.8)	1.6186	2.8820	1.6541	2.9492	***1.4400***	***2.5744***	1.5597	2.7558	1.4487	2.5779	***1.2445***	***2.1201***
Fig. 11	32.3044	31.4755	31.3102	31.4701	34.0872	***31.8521***	32.5929	31.7561	32.1448	32.5350	***36.1269***	***33.5147***
(No. 19)	1.6422	1.9396	1.6734	1.8828	***1.4397***	***1.6772***	1.5637	1.8726	1.4981	1.7634	***1.2793***	***1.5336***

regions of demosaiced image will still exist when utilizing Gunturk's and Lu's demosaicing methods. On the contrary, color artifacts of demosaiced images obtained from the proposed EADA interpolation step are mainly converged in edge regions. The discussion of color-difference model in Section 2 shows that the color artifacts in high-frequency regions will be reduced effectively by linear low-pass filtering in the color-difference space. Therefore, the proposed post-processing method provides superior improvement for the demosaiced images obtained from the interpolation step. More specifically, EADA provides the fewest color artifacts for smooth regions in interpolation step and reduces the residual color artifacts around edge regions effectively in post-processing step. This can be seen in Table 3 and by visually comparing Fig. 9(d) with Fig. 9(g). Similar results can also be observed in the test results presented Fig. 10.

The demosaiced results shown in Fig. 11 evaluate the performance of EADA in edge regions and fine textures. Fig. 11(a) shows a zoom-in of the test image No. 19. In this scene, the fence region contains many long edges and high-frequency detail of the image. These features usefully challenge the performance of demosaicing methods. Figs. 11(b)–(d) are, respectively the demosaiced results obtained from Gunturk's, Lu's and EADA interpolation methods. From visual comparison, we can observe that the Gunturk's and Lu's interpolation methods induce more color artifacts in the fence region than EADA does. Moreover, as shown in Figs. 11(e) and (f), the color artifacts in the demosaiced result still remain in the fence region when using Lu's post-processing method. However, by visually comparing Figs. 11(d) with (g), it is clear that the color artifacts in the demosaiced result can be effectively reduced by using the proposed post-processing method. Note in Table 3 that the proposed algorithm provides the best PSNR and ΔE^*_{ab} measures not only in the interpolation step, but also in the post-processing step for this case. The gap between proposed and other methods can be as large as 3.57 dB (PSNR) and 0.22 ΔE^*_{ab} units in smooth region, 0.98 dB (PSNR) and 0.23 ΔE^*_{ab} units in edge region. These experimental results validate that EADA not only effectively reduces the color artifacts in edge regions and fine textures,

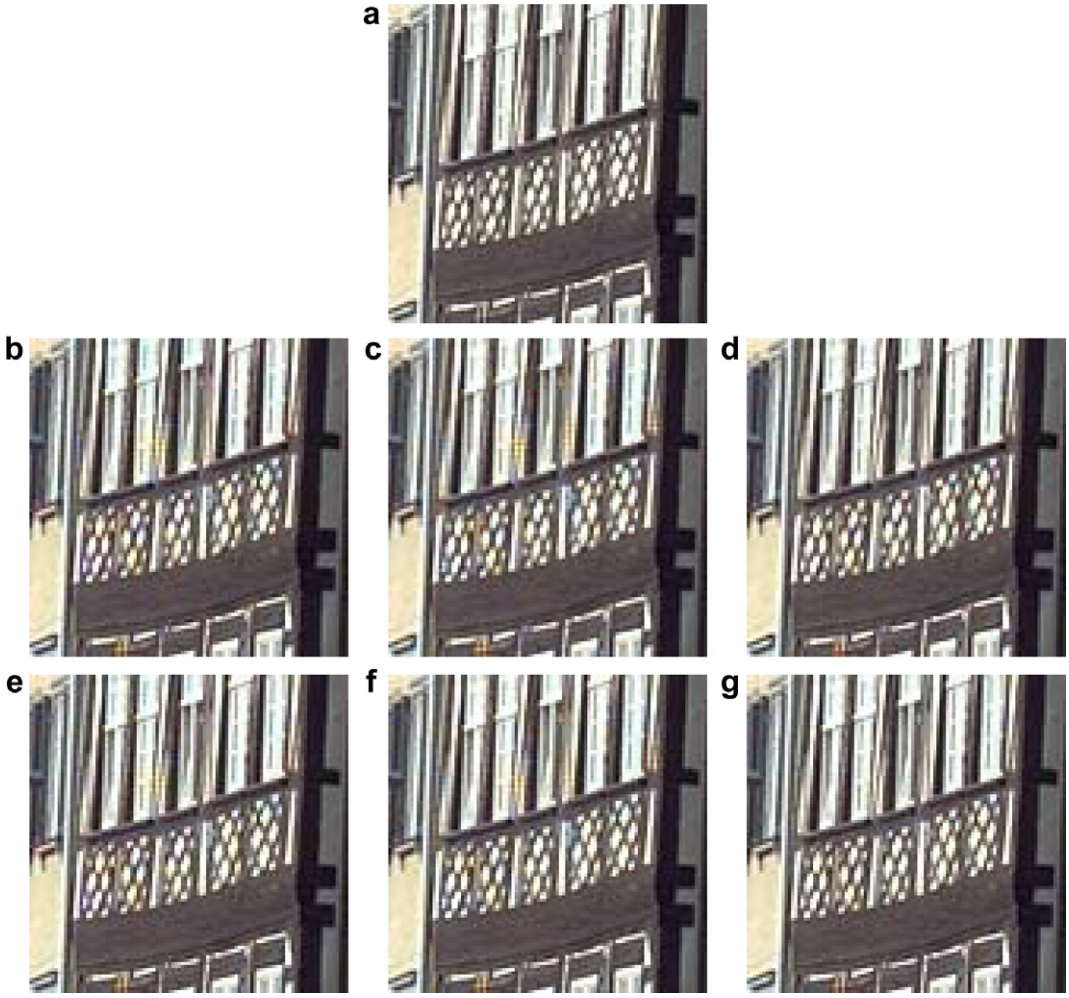

Fig. 10. Zoom-in demosaicing test results of test image No. 8: (a) original picture, (b) demosaiced result of Gunturk's method, (c) Lu's interpolation result, (d) the proposed EADA interpolation result, (e) Gunturk's method with Lu's post-processing result, (f) Lu's post-processing result, and (g) the proposed EADA post-processing result.

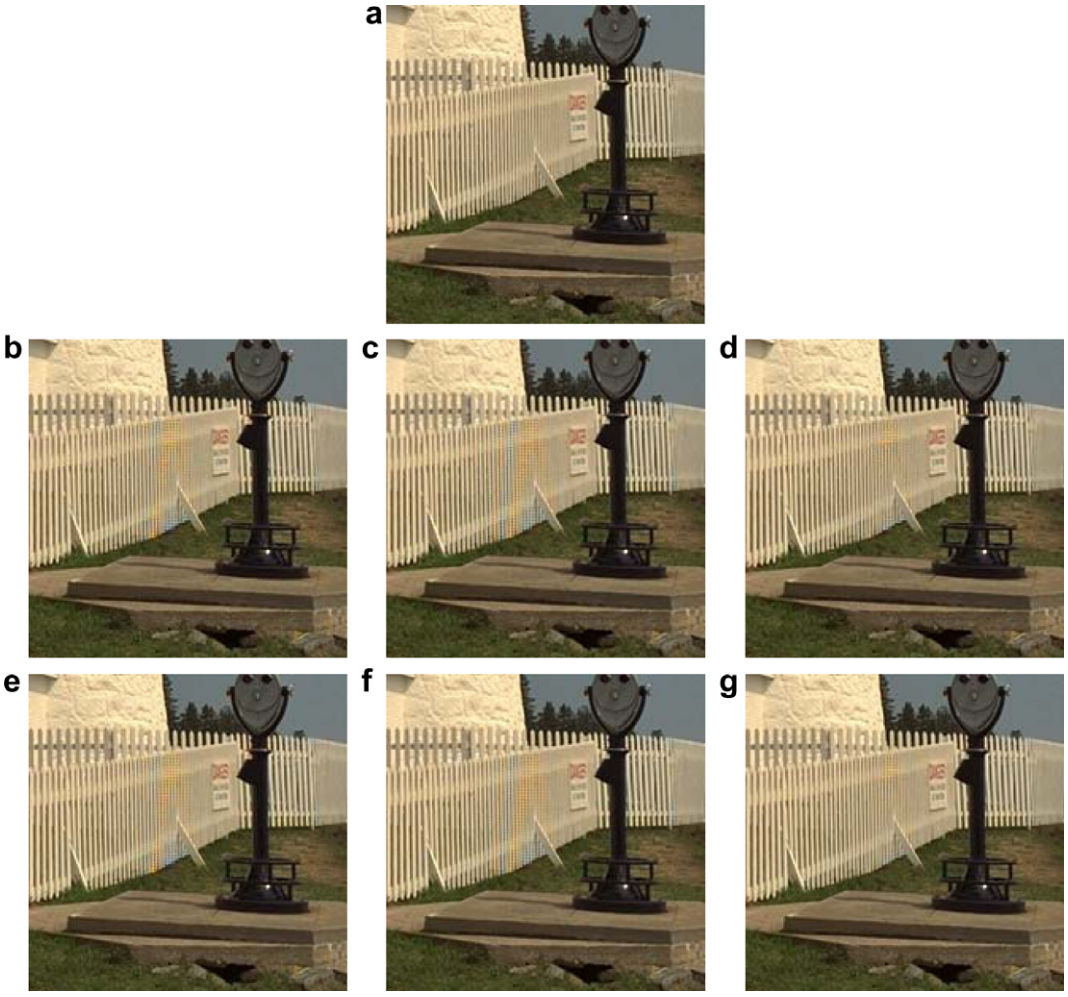

Fig. 11. Zoom-in demosaicing test results of test image No. 19: (a) original picture, (b) demosaiced result of Gunturk's method, (c) Lu's interpolation result, (d) the proposed EADA interpolation result, (e) Gunturk's method with Lu's post-processing result, (f) Lu's post-processing result, and (g) the proposed EADA post-processing result.

but also gives superior performance compared with the two well-known methods.

6. Conclusion and future work

This paper presents a novel edge-adaptive CFA demosaicing algorithm based on a color-difference approach. The proposed EADA algorithm effectively reduces color artifacts in both smooth and edge regions of demosaiced images. The proposed algorithm can be combined with any existing image interpolation method for reconstructing the green channel. A new adaptive interpolation method is then presented using a nonlinear procedure for choosing the direction of interpolation to reconstruct the green channel from CFA samples. The performance of the proposed method has been compared with two recent published demosaicing methods, Gunturk's and Lu's methods. Experimental results show that the proposed method not only outperforms both of them in PSNR (dB) and ΔE^*_{ab} measures, but also gives superior demosaiced fidelity in visual comparison with other methods. Future research directions will focus on developing the single-plane reconstruction algorithms to reconstruct the green channel with minimum interpolation error.

Acknowledgements

The authors would like to thank Prof. B.K. Gunturk of Louisiana State University, USA and Dr. Yap-Peng Tan of Nanyang Technological University, Singapore for providing us their CFA demosaicing programs. This work was partly supported by the National Science Council of Taiwan, ROC under Grant NSC 92-2213-E-009-007.

References

[1] D.R. Cok, Reconstruction of CCD images using template matching, in: Proc. IS&T, 47th Annual Conf./ICPS (1994) 380–385.
[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, 1987.
[3] W.T. Freeman, Method and apparatus for reconstructing missing color samples, U.S. Patent 4,774,565, 1988.

[4] J.E. Adams, Interactions between color plane interpolation and other image processing functions in electronic photography, Proc. SPIE Cameras Syst. Electronic Photogr. Sci. Imag. 2416 (1995) 144–151.
[5] J. Hamilton, J. Adams, Adaptive color plane interpolation in single sensor color electronic camera, U.S. Patent 5,629,734, 1997.
[6] J.E. Adams, Design of practical color filter array interpolation algorithms for digital cameras, Proc. SPIE Real Time Imag. II 3028 (1997) 117–125.
[7] E. Chang, S. Cheung, D.Y. Pan, Color filter array recovery using a threshold-based variable number of gradients, Proc. SPIE Sensors, Cameras, Appl. Digital Photogr. 3650 (1999) 36–43.
[8] R.H. Hibbard, Apparatus and method for adaptively interpolation a full color image utilizing luminance gradients, U.S. Patent 5,382,976, 1996.
[9] R. Kimmel, Demosaicing: image reconstruction from color CCD samples, IEEE Trans. Image Process. 8 (9) (1999) 1221–1228.
[10] X. Li, M.T. Orchard, New edge-directed interpolation, IEEE Trans. Image Process. 10 (10) (2001) 1521–1527.
[11] D.D. Muresan. T.W. Parks, Optimal recovery approach to image interpolation, in: Proc. IEEE. Int. Conf. Image Process., vol. 3, 2001, pp. 848–851.
[12] K. Hirakawa, T.W. Parks, Adaptive homogeneity-directed demosaicing algorithm, in: Proc. IEEE Int. Conf. Image Process., vol. 1, 2003, pp. 669–672.
[13] B.K. Gunturk, Y. Altunbasak, R.M. Mersereau, Color plane interpolation using alternating projections, IEEE Trans. Image Process. 11 (9) (2002) 997–1013.
[14] W. Lu, Y.P. Tan, Color filter array demosaicing: new method and performance measures, IEEE Trans. Image Process. 12 (10) (2003) 1194–1210.
[15] L. Chang, Y.P. Tan, Effective use of spatial and spectral correlations for color filter array demosaicing, IEEE Trans. Consumer Electronics 50 (1) (2004) 355–365.
[16] M. Mahy, E. Van Eyckden, O. Oosterlinck, Evaluation of uniform color spaces developed after the adoption of CIELAB and CIELUV, Color Res. Appl. 19 (2) (1994) 105–121.
[17] B. Bayer, Color imaging array, U.S. Patent 3,971,065, 1976.
[18] R.C. Gonzalez, R.E. Woods, Digital Image Processing, second ed., Prentice-Hall, 2002.
[19] D.D. Muresan, T.W. Parks, Adaptively quadratic (AQua) image interpolation, IEEE Trans. Image Process. 13 (5) (2004) 690–698.