Journal of Electronic Imaging

JElectronicImaging.org

# Joint demosaicking and zooming using moderate spectral correlation and consistent edge map

Dengwen Zhou
Weiming Dong
Wengang Chen

SPIE.
IS&T
imaging.org

# Joint demosaicking and zooming using moderate spectral correlation and consistent edge map

**Dengwen Zhou,[a,*] Weiming Dong,[b] and Wengang Chen[a]**
[a]North China Electric Power University, School of Control and Computer Engineering, 2 Beinong Road, Changping District, Beijing 102206, China
[b]LIAMA-NLPR, CAS Institute of Automation, Haidian District, Beijing 100190, China

**Abstract.** The recently published joint demosaicking and zooming algorithms for single-sensor digital cameras all overfit the popular Kodak test images, which have been found to have higher spectral correlation than typical color images. Their performance perhaps significantly degrades on other datasets, such as the McMaster test images, which have weak spectral correlation. A new joint demosaicking and zooming algorithm is proposed for the Bayer color filter array (CFA) pattern, in which the edge direction information (edge map) extracted from the raw CFA data is consistently used in demosaicking and zooming. It also moderately utilizes the spectral correlation between color planes. The experimental results confirm that the proposed algorithm produces an excellent performance on both the Kodak and McMaster datasets in terms of both subjective and objective measures. Our algorithm also has high computational efficiency. It provides a better tradeoff among adaptability, performance, and computational cost compared to the existing algorithms. *© 2014 SPIE and IS&T* [DOI: 10.1117/1.JEI.23.4.043010]

Keywords: color filter array; demosaicking; zooming; directional interpolation; Bayer pattern.

Paper 13674 received Dec. 1, 2013; revised manuscript received Jun. 6, 2014; accepted for publication Jun. 20, 2014; published online Jul. 21, 2014.

## 1 Introduction

Most digital color cameras use a single sensor with a color filter array (CFA), such as the Bayer CFA pattern (Fig. 1), which allows only a red (R), green (G), or blue (B) color value to be measured at each pixel. The missing two color values at each pixel must be estimated. This estimation process is referred to as demosaicking.[1,2] After the demosaicking procedure, image zooming (i.e., enlargement) is one of the most commonly performed processing operations in a digital camera because the cost of digital cameras increases rapidly with optical zooming capabilities.[3] To produce a full-color zooming result from a CFA image, the typical approach is that the CFA image is first demosaicked and then zoomed. Lukac et al.[3,4] provided an alternative approach, i.e., the CFA image is first zoomed and then demosaicked. The advantage of those two approaches is that demosaicking and zooming processes are independent, and all existing demosaicking and zooming algorithms can be directly utilized. However, when zooming and demosaicking are performed separately, the information available on the raw sensor data cannot be utilized consistently and efficiently.[5] Some researchers have demonstrated that the third approach where demosaicking and zooming are implemented simultaneously can improve the quality of the resulting images.[5–7]

A common assumption in demosaicking is that there is high spectral correlation between color planes.[1,2] Almost all existing color demosaicking algorithms exploit the correlation to interpolate the missing color samples, and the popular Kodak set of test images[8] is commonly used to benchmark demosaicking performance. However, some researchers[9,10] have noted that the Kodak test images have higher spectral correlation than typical color images. The demosaicking algorithms overemphasizing the spectral correlation are prone to large interpolation errors and can produce highly visible color artifacts. Zhang et al.[11] proposed a local directional interpolation and nonlocal adaptive thresholding (LDI-NAT) demosaicking algorithm and tried to remedy the above problem. Their algorithm first obtains an initial demosaicking result by local directional interpolation and fusion, and then searches the pixels similar to each estimated pixel in a larger neighborhood to enhance the estimated result. LDI-NAT has been evaluated using the McMaster dataset,[12] which has weak spectral correlation and highly saturated colors, and outperforms many state-of-the-art demosaicking algorithms. However, we found that LDI-NAT is inferior to the other state-of-the-art demosaicking algorithms over the Kodak dataset.

In this paper, we propose a new joint demosaicking and zooming algorithm based on the commonly used Bayer CFA pattern. In our algorithm, the edge direction information (edge map) extracted from the raw CFA data is consistently used in demosaicking and zooming. It also moderately utilizes the spectral correlation between color planes. Its backbone is the reconstruction of the G plane, and then the reconstructed G plane is used for the reconstruction of R and B planes by color difference planes. The proposed algorithm was evaluated over both the Kodak and McMaster datasets, and obviously outperforms the other similar algorithms, i.e., the ones proposed by Chung and Chan,[5] Zhang and Zhang,[6] and Chung et al.[7] It is also superior to the zooming algorithms based on LDI-NAT demosaicking.[11]

The remainder of this paper is organized as follows. In Sec. 2, the details of the proposed joint demosaicking and zooming algorithm are described. Section 3 is the experimental results and discussions of the algorithms under comparison. Finally, a conclusion is given in Sec. 4.

*Address all correspondence to: Dengwen Zhou, E-mail: zdw@ncepu.edu.cn

0091-3286/2014/$25.00 © 2014 SPIE and IS&T

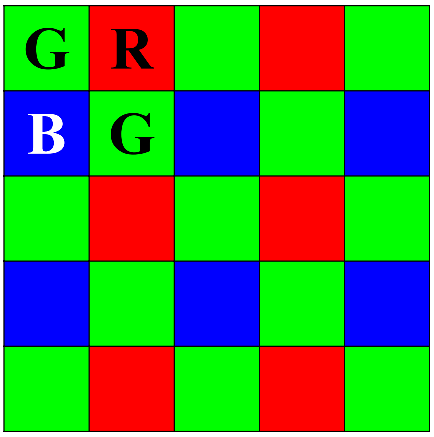


**Fig. 1** Bayer color filter array (CFA) pattern.

## 2 Proposed Joint Demosaicking and Zooming Algorithm

In this section, the details of our algorithm are given. The joint demosaicking and zooming algorithms of Refs. 5, 6, and 7 all use a zooming factor of 2. Without loss of generality, a zooming factor of 2 will also be used in this paper. That is, assume that a CFA image of size $M \times N$ has to be enlarged to a zoomed full-color image of size $2M \times 2N$. The demosaicking process of our algorithm, including the interpolation and refinement of G plane and color difference planes, is similar to that of Ref. 13, and more detailed information is provided therein.

### 2.1 *Interpolation of Green Plane*

In the Bayer pattern as in Fig. 1, the sampling frequency of the G plane is twice as high as that of R and B color planes, and consequently, the G plane can be interpolated more accurately.[1,13] We first consider the interpolation of the G plane, which is an improved version of the one described in Ref. 14 and is comprised of two steps: a directional interpolation step and a decision step.

#### 2.1.1 *Directional interpolation step*

This step interpolates the missing G values using the famous adaptive color plane interpolator proposed by Hamilton and Adams[15] in the horizontal and vertical directions, respectively. Estimating the missing G value at an R sampling location will be considered [Fig. 2(a)]. The interpolation at a B sampling location [Fig. 2(b)] can be similarly carried out. In Fig. 2(a), the interpolations $G^H$ and $G^V$ of the central missing G sample in the horizontal and vertical directions are as follows:

$$\begin{cases} G^H_{i,j} = \frac{G_{i,j-1}+G_{i,j+1}}{2} + \frac{2R_{i,j}-R_{i,j-2}-R_{i,j+2}}{2} \\ G^V_{i,j} = \frac{G_{i-1,j}+G_{i+1,j}}{2} + \frac{2R_{i,j}-R_{i-2,j}-R_{i+2,j}}{2} \end{cases}. \tag{1}$$

#### 2.1.2 *Decision step*

Once the G plane has been interpolated along both horizontal and vertical directions, two G images can be produced. The selection of a better direction from the two for every sample is required. Every direction can be determined based on the aggregated gradients of color differences in a local neighborhood. Therefore, two color difference images are computed.

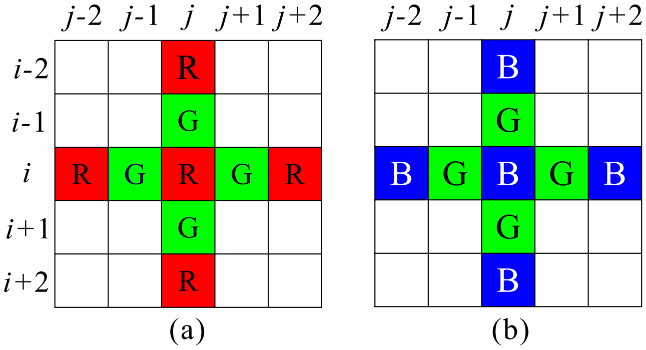


**Fig. 2** Reference CFA samples having centers at (a) red and (b) blue samples.

$$\begin{cases} C_H(i,j) = \begin{cases} R_{i,j} - G^H_{i,j}, & \text{if pixel}(i,j)\text{is a red location} \\ B_{i,j} - G^H_{i,j}, & \text{if pixel}(i,j)\text{is a blue location} \end{cases} \\ C_V(i,j) = \begin{cases} R_{i,j} - G^V_{i,j}, & \text{if pixel}(i,j)\text{is a red location} \\ B_{i,j} - G^V_{i,j}, & \text{if pixel}(i,j)\text{is a blue location} \end{cases} \end{cases}$$

where $i$ and $j$ indicate the row and column of pixel $(i,j)$. Next, the gradients of $C_H$ and $C_V$ are calculated as $D_H(i,j) = |C_H(i,j) - C_H(i,j+2)|$ and $D_V(i,j) = |C_V(i,j) - C_V(i+2,j)|$. Subsequently, two classifiers $\delta_H(i,j)$ and $\delta_V(i,j)$ are defined as the sum of the gradients $D_H$ and $D_V$ in a $5 \times 5$ neighborhood centered at pixel $(i,j)$. They give an estimate of the local variation of the color differences along the horizontal and vertical directions, respectively, and can be used to estimate the edge direction. For all R and B sampling locations, the missing G values are estimated using the following criterion:

$$\begin{cases} \text{if } [1+\delta_V(i,j)]/[1+\delta_H(i,j)] > T_1 \\ G_{i,j} = G^H_{i,j} \\ \text{else if } [1+\delta_H(i,j)]/[1+\delta_V(i,j)] > T_1 \\ G_{i,j} = G^V_{i,j} \\ \text{else} \\ G_{i,j} = (w_1 * G^H_{i,j} + w_2 * G^V_{i,j})/(w_1 + w_2) \\ \text{end} \end{cases}, \tag{2}$$

where 1 is added to the classifiers to avoid division by zero. $T_1$ is a threshold and needs to be carefully chosen because the ratio between the two classifiers suggests the strength of a horizontal or vertical edge. The greater the value of $T_1$, the more the region is considered to be smooth. The above criterion indicates that if the missing G value $G_{i,j}$ is on a horizontal strong edge (or on a vertical strong edge), then $G_{i,j} = G^H_{i,j}$ (or $G_{i,j} = G^V_{i,j}$), otherwise $G_{i,j}$ is in a weak edge (or textured) region and takes a weighted average of $G^H_{i,j}$ and $G^V_{i,j}$, whose contributions to $G_{i,j}$ should be proportional to the strength of the horizontal and vertical edges. Since the strength of a directional edge is inversely proportional to its directional gradient,[3,14] the weights $w_1$ and $w_2$ can be considered as the reciprocal of the directional gradients. We compute horizontal and vertical gradients using the exact same method as described in Ref. 15. The calculation of the weights $w_1$ and $w_2$ has two cases where the location $(i,j)$ of the missing G sample is an R or B location. Here, calculating $w_1$ and $w_2$ at an R location will be considered [Fig. 2(a)].

The calculation at a B location can be similarly carried out. The weights $w_1$ and $w_2$ are computed as follows:

$$\begin{cases} w_1 = 1/(1 + |G_{i,j+1} - G_{i,j-1}| + |2R_{i,j} - R_{i,j-2} - R_{i,j+2}|) \\ w_2 = 1/(1 + |G_{i+1,j} - G_{i-1,j}| + |2R_{i,j} - R_{i-2,j} - R_{i+2,j}|) \end{cases}, \tag{3}$$

where 1 is also added to the denominators to avoid division by zero.

The decided edge direction information using Eq. (2) at the missing G sample locations is recorded in the edge map $DM$. Each missing G sample may locate on a horizontal (or vertical) strong edge or in a weak edge (or textured) region. We set $DM(i,j)$ as

$$DM(i,j) = \begin{cases} 1, & \text{if pixel}(i,j)\text{is on a vertical strong edge} \\ 2, & \text{if pixel}(i,j)\text{is on a horizontal strong edge} \\ 3, & \text{if pixel}(i,j)\text{is in a weak edge or textured region} \\ 0, & \text{otherwise} \end{cases}. \tag{4}$$

The edge map $DM$ will continue to be used in the subsequent zooming of the G plane and color difference planes.

### 2.2 *Interpolation of Color Difference Planes R-G and B-G*

After the G plane is fully populated, it can be used to assist the subsequent interpolation of color difference planes R-G and B-G, because the estimated G values can be considered to be known. It is well known that the spectral correlation between color planes can be exploited to improve demosaicking results. The most common spectral correlation model is probably the color difference model, which has been extensively used in the demosaicking community.[9] Color difference planes are smoother than primary color planes,[6,13] and interpolation tends to provide better results for smoother images.[1] Therefore, it is preferable to estimate and enlarge R and B planes by color difference planes R-G and B-G.

If $(i,j)$ is an R (or a B) sampling location in the CFA image, $R_{i,j} - G_{i,j}$ (or $B_{i,j} - G_{i,j}$) can be directly computed. Otherwise, if $(i,j)$ is a G sampling location, $R_{i,j} - G_{i,j}$ and $B_{i,j} - G_{i,j}$ must be estimated because the R and B samples are missing. In addition, if $(i,j)$ is a B (or an R) sampling location, $R_{i,j} - G_{i,j}$ (or $B_{i,j} - G_{i,j}$) must also be estimated because the R (or B) sample is missing. Here, only the color difference B-G will be estimated, because the estimation of R-G can be carried out analogously.

Considering that $(i,j)$ is a G or an R sampling location, there are three different cases for estimating $B_{i,j} - G_{i,j}$, as shown in Fig. 3. Because each R sampling location has four adjacent known B samples in two diagonal directions [Fig. 3(c)], the color difference values at the R sampling locations will first be estimated. Letting $KB$ and $KR$ denote color differences B-G and R-G, respectively, i.e., $KB_{i,j} = B_{i,j} - G_{i,j}$ and $KR_{i,j} = R_{i,j} - G_{i,j}$, the central position $KB_{i,j}$ in Fig. 3(c) can be estimated by the weighted average of the four diagonal adjacent color difference values.

$$KB_{i,j} = \frac{w_{i-1,j-1}KB_{i-1,j-1} + w_{i-1,j+1}KB_{i-1,j+1} + w_{i+1,j-1}KB_{i+1,j-1} + w_{i+1,j+1}KB_{i+1,j+1}}{w_{i-1,j-1} + w_{i-1,j+1} + w_{i+1,j-1} + w_{i+1,j+1}}, \tag{5}$$

where the computation of $KB_{x,y}$ at $(x,y)$ is straightforward, and the weight $w_{x,y}$ is computed by the reciprocal of the gradient of the four diagonal adjacent color difference values [see Eq. (6)]. The idea supporting the computation is similar to the one in Eq. (3).

$$\begin{cases} w_{i-1,j-1} = \frac{1}{1+|KB_{i+1,j+1}-KB_{i-1,j-1}|+c|KB_{i-1,j-1}-KB_{i-3,j-3}|} \\ w_{i-1,j+1} = \frac{1}{1+|KB_{i+1,j-1}-KB_{i-1,j+1}|+c|KB_{i-1,j+1}-KB_{i-3,j+3}|} \\ w_{i+1,j-1} = \frac{1}{1+|KB_{i-1,j+1}-KB_{i+1,j-1}|+c|KB_{i+1,j-1}-KB_{i+3,j-3}|} \\ w_{i+1,j+1} = \frac{1}{1+|KB_{i-1,j-1}-KB_{i+1,j+1}|+c|KB_{i+1,j+1}-KB_{i+3,j+3}|} \end{cases}, \tag{6}$$

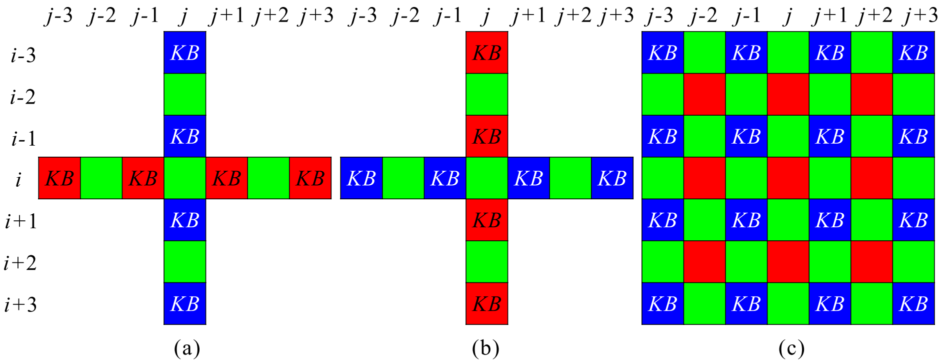


**Fig. 3** Three cases of central missing blue/green color differences. (a) and (b) Centers at green sampling locations. (c) Center at a red sampling location.

where $c$ is a constant factor adjusting the weighting effect and 1 is added to the denominators to avoid division by zero.

Next, the color difference values at the G sampling positions will be estimated. Every G sampling position has two adjacent known B samples in the horizontal direction [Fig. 3(a)] or in the vertical direction [Fig. 3(b)]. After the color difference values of B-G at the R sampling positions are estimated, every G sampling position also has two adjacent estimated color difference values of B-G in the horizontal direction [Fig. 3(b)] or in the vertical direction [Fig. 3(a)].

The central position $KB_{i,j}$ in Fig. 3(a) or 3(b) can also be estimated similarly to Eq. (5) using the four adjacent color difference values in the horizontal and vertical directions instead of the diagonal adjacent color difference values.

$$KB_{i,j} = \frac{w_{i-1,j}KB_{i-1,j} + w_{i,j+1}KB_{i,j+1} + w_{i,j-1}KB_{i,j-1} + w_{i+1,j}KB_{i+1,j}}{w_{i-1,j} + w_{i,j+1} + w_{i,j-1} + w_{i+1,j}}, \tag{7}$$

where the weight $w_{x,y}$ at $(x, y)$ is also computed similarly to Eq. (6)

$$\begin{cases} w_{i-1,j} = \frac{1}{1+|KB_{i+1,j}-KB_{i-1,j}|+c|KB_{i-1,j}-KB_{i-3,j}|} \\ w_{i,j+1} = \frac{1}{1+|KB_{i,j-1}-KB_{i,j+1}|+c|KB_{i,j+1}-KB_{i,j+3}|} \\ w_{i,j-1} = \frac{1}{1+|KB_{i,j+1}-KB_{i,j-1}|+c|KB_{i,j-1}-KB_{i,j-3}|} \\ w_{i+1,j} = \frac{1}{1+|KB_{i-1,j}-KB_{i+1,j}|+c|KB_{i+1,j}-KB_{i+3,j}|} \end{cases}. \tag{8}$$

## 2.3 Refinement

After the initial interpolation step, the G plane and two color difference planes are fully populated and their estimates can be further refined. With the aid of the populated color difference planes, the G plane can be refined. Then the refined G plane is used to refine the color difference planes again. Only the estimated G values in the G plane will be refined, and the original CFA-sampled G values are not altered. Similarly, R-G (or B-G) will be refined only at the G and B (or G and R) sampling positions.

### 2.3.1 Refinement of green plane

Refining the G value at a B sampling position will be considered. The refinement at an R sampling position can be similarly carried out. Let us refine the G sample $G_{i,j}$ at a B sampling position [Fig. 4(a)], $G_{i,j} = B_{i,j} - KB_{i,j}$, where $KB_{i,j}$ can be computed using Eq. (7), and the weights computing $KB_{i,j}$ in Eq. (7) can be computed using Eq. (8).

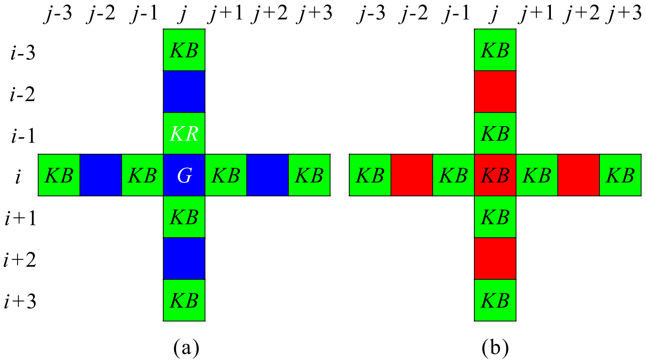


**Fig. 4** Central green sample and blue/green color differences need to be refined. (a) Refinement of green sample at a blue sampling location. (b) Refinement of blue/green color difference at a red sampling location.

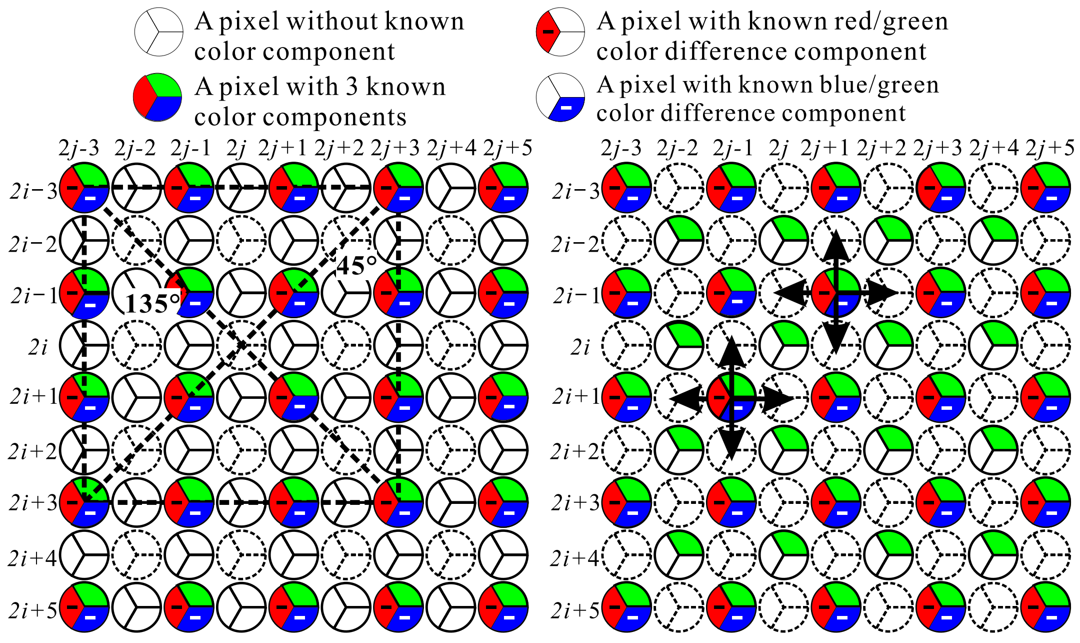
### 2.3.2 Refinement of color difference planes

With the refined G plane, color difference planes B-G and R-G can be updated and further refined. Since refining R-G is similar to refining B-G, only the refinement of B-G is considered here. B-G needs to be refined at the G or R sampling position. With the initial estimated B-G plane, both G and R sampling positions have four adjacent values of B-G in the horizontal and vertical directions. Just as with the refinement of the G plane, the refinement of B-G is also performed by Eqs. (7) and (8) for every G and R sampling position [please refer to Fig. 4(b) or Figs. 3(a) and 3(b)].

## 2.4 Enlargement of Green Plane

After the G and color difference planes have been interpolated, we can also obtain the edge map $DM$, which includes the edge direction information at every missing G sample position in the raw CFA image. Now the demosaicked G plane is expanded to a twice enlarged grid, and then the missing G values [please refer to Fig. 5(a)] are filled. The edge map $DM$ is also expanded according to the grid in order to record the new estimated edge direction information. (Notice that the enlarged $DM$ is still notated as $DM$.) We can see from Fig. 5(a) that, for the two times zoom, three quarters of the G values are missing and need to be estimated. The missing G values are interpolated in two passes. The G values of the pixels marked by the dashed lines in Fig. 5(a) will be interpolated in the first pass. The G values of the other missing pixels [please refer to the pixels marked by the dashed lines in Fig. 5(b)] will be interpolated in the second pass. For every G sample to be estimated in the first pass, there are the G samples of four diagonal adjacent pixels to be known as illustrated in Fig. 5(a). To determine its interpolation direction, we estimate the strength of the edges in 45 and 135 deg diagonal directions in terms of the gradients in the two directions, respectively. Since a local gradient is greater across an edge than along it, it can indicate the strength of a local edge. If a G sample is missing on a strong edge, we estimate it by the bicubic interpolation proposed by Keys[16] along the estimated strong edge. If a G sample is missing on a weak edge (or textured) region, it is estimated by combining the two orthogonal directional bicubic interpolation results. At the same time, every new estimated edge direction is also recorded in the edge map $DM$, and the newly added $DM(i, j)$ is exactly the same as the one defined in Eq. (4).

According to the known G values in a $7 \times 7$ neighborhood referring to the pixels in the dashed square of Fig. 5(a), the gradients $\delta_1$ and $\delta_2$ at location $(2i, 2j)$ are computed as


A pixel without known color component
A pixel with 3 known color components
A pixel with known red/green color difference component
A pixel with known blue/green color difference component
2j-3 2j-2 2j-1 2j 2j+1 2j+2 2j+3 2j+4 2j+5
2i−3
2i−2
2i−1
2i
2i+1
2i+2
2i+3
2i+4
2i+5
135°
45°

(a) (b)

**Fig. 5** Illustration of image zooming.

$$
\begin{cases}
\delta_1 = \sum\limits_{m=3,\pm 1} \sum\limits_{n=3,\pm 1} |G(2i+m, 2j-n) - G(2i+m-2, 2j-n+2)| (45\ \text{deg diagonal}) \\
\delta_2 = \sum\limits_{m=3,\pm 1} \sum\limits_{n=3,\pm 1} |G(2i+m, 2j+n) - G(2i+m-2, 2j+n-2)| (135\ \text{deg diagonal})
\end{cases}, \tag{9}
$$

where $G(x,y)$ is the known G value located at $(x,y)$ in the enlarged grid of the G plane. We can estimate the edge direction similar to Eq. (2) using the ratio of the two orthogonal directional gradients.

$$
\begin{cases}
\text{if } (1+\delta_1)/(1+\delta_2) > T_2 \\
\text{pixel}(2i,2j)\text{is on a } 135\ \text{deg strong edge,} \\
\text{set } DM(2i,2j) = 1 \\
\text{else if } (1+\delta_2)/(1+\delta_1) > T_2 \\
\text{pixel}(2i,2j)\text{is on a } 45\ \text{deg strong edge,} \\
\text{set } DM(2i,2j) = 2 \\
\text{else} \\
\text{pixel}(2i,2j)\text{is in a weak edge or textured region,} \\
\text{set } DM(2i,2j) = 3 \\
\text{end}
\end{cases}, \tag{10}
$$

where 1 is added to the gradients to avoid division by zero and $T_2$ is a threshold parameter, which is a key factor for the decision of whether a pixel is on a strong edge. The estimated edge direction information is recorded in the edge map $DM$ again. Assuming that the two orthogonal directional bicubic interpolation values at location $(2i, 2j)$ are $p_1$ (45 deg diagonal directional) and $p_2$ (135 deg diagonal directional), $p_1$ and $p_2$ can easily be computed using the bicubic interpolation filter $[-1, 9, 9, -1]/16$.[16,17] Again assuming that pixel $(2i, 2j)$ is in a weak edge (or textured) region, the missing G value $p$ of pixel $(2i, 2j)$ can be estimated as

$$
p = (w_1 p_1 + w_2 p_2)/(w_1 + w_2), \tag{11}
$$

where the weights $w_1$ and $w_2$ combined with $p_1$ and $p_2$ are also computed similarly to Eq. (3).

$$
\begin{cases}
w_1 = \frac{1}{1+\delta_1^k} \\
w_2 = \frac{1}{1+\delta_2^k}
\end{cases}, \tag{12}
$$

where 1 is added to the denominators to avoid division by zero and $k$ is an exponent parameter adjusting the weighting effect.

For every pixel to be estimated in the second pass, i.e., every pixel marked by the dashed lines in Fig. 5(b), four immediately adjacent G values in the horizontal and vertical directions are known (in which two immediately adjacent G values are estimated in the first pass). Further, one of the four immediately adjacent pixels in the horizontal and vertical directions [if located at $(\tilde{i}, \tilde{j})$] includes the edge direction information, which has been estimated during the demosaicking of the G plane and has been recorded in the edge map $DM$. At this point, the missing G value $p$ [if located at $(i, j)$] can be estimated with the help of the edge map $DM$.

$$\begin{cases}\text{if } DM(\tilde{i},\tilde{j})\text{represents the horizontal direction}\\ p=p_1\\ \text{else if } DM(\tilde{i},\tilde{j})\text{represents the vertical direction}\\ p=p_2\\ \text{else}\\ \text{pixel}(i,j)\text{is in a weak edge or textured region,}\\ \quad p\,\text{is estimated using}\,\text{Eq.}(11)\\ \text{end}\end{cases}, \tag{13}$$

where $p_1$ and $p_2$ represent the bicubic interpolation values of the missing G sample at $(i,j)$ in the horizontal and vertical directions, respectively. If pixel $(i,j)$ is in a weak edge (or textured) region, the missing G sample $p$ is still computed using Eq. (11), and the weights combining $p_1$ and $p_2$ in Eq. (11) are still computed using Eq. (12). The gradients $\delta_1$ and $\delta_2$ in Eq. (12) can be computed similarly to Eq. (9) using the known G values in the horizontal and vertical directions instead of the diagonal directions. That is,

$$\begin{cases}\delta_1=\sum\limits_{m=\pm1}\sum\limits_{n=0,2}|G(i+m,j-n)-G(i+m,j-n+2)|+\sum\limits_{m=0,\pm2}|G(i+m,j-1)-G(i+m,j+1)|(\text{horizontal})\\ \delta_2=\sum\limits_{m=0,2}\sum\limits_{n=\pm1}|G(i-m,j+n)-G(i-m+2,j+n)|+\sum\limits_{m=0,\pm2}|G(i-1,j+n)-G(i+1,j+n)|(\text{vertical})\end{cases}. \tag{14}$$

## 2.5 *Enlargement of Color Difference Planes R-G and B-G*

After the G plane has been enlarged, color difference planes R-G and B-G will be enlarged. The enlarged R and B planes can be obtained by adding the enlarged G plane to color difference planes R-G and B-G. Two of the most well-known and most widely used traditional interpolation methods are the bilinear and bicubic.[17,18] Generally, using bicubic interpolation can achieve a slightly better performance than using bilinear, although at the expense of greater computation time. Considering that color difference planes are much smoother than original color planes,[6,13] bilinear interpolation can reach good enough results. We estimate the missing color difference values only by the simple directional bilinear interpolation with the help of the edge map $DM$. We still refer to Fig. 5; here the missing color difference values of the pixels marked by the dashed lines will be estimated. As the enlargement of the G plane, the missing color difference values are also interpolated in two passes.

The color difference values of the pixels marked by the dashed lines in Fig. 5(a) are interpolated in the first pass. If the missing color difference value $p$ is at $(i,j)$, and $DM(i,j)$ represents the 45 deg diagonal direction [$DM(i,j)$ has been computed during the process of the G plane zooming], $p$ is estimated as the mean of the two color difference values that are immediately adjacent to $(i,j)$ in the 45 deg diagonal direction. If $DM(i,j)$ represents the 135 deg diagonal direction, $p$ can be similarly estimated. If $DM(i,j)$ represents the weak edge (or textured) region, $p$ will be estimated as the mean of the four color difference values that are immediately adjacent to $(i,j)$ in the 45 and 135 deg diagonal directions.

The color difference values of the other missing pixels marked by the dashed lines in Fig. 5(b) are interpolated in the second pass. Similarly, if the missing color difference value $p$ is at $(i,j)$, $(\tilde{i},\tilde{j})$ represents the position of the missing G sample in the raw CFA image, which is immediately adjacent to $(i,j)$ in the horizontal or vertical direction. $DM(\tilde{i},\tilde{j})$ has been computed during the demosaicking of the G plane. If $DM(\tilde{i},\tilde{j})$ represents the horizontal direction, $p$ is estimated as the mean of the two color difference values that are immediately adjacent to $(i,j)$ in the horizontal direction. If $DM(\tilde{i},\tilde{j})$ represents the vertical direction, $p$ can be estimated similarly. If $DM(\tilde{i},\tilde{j})$ represents the weak edge (or textured) region, $p$ will be estimated as the mean of the four color difference values that are immediately adjacent to $(i,j)$ in the horizontal and vertical directions.

## 2.6 *Summary of the Algorithm*

Our algorithm can be outlined using the flow chart in Fig. 6. Our proposed workflow is as follows:

1. Extract the edge direction information in the missing G sample positions, which is recorded in the edge map $DM$ (i.e., a matrix indicating the edge direction information of every pixel), and recover the G plane with the aid of the edge map $DM$ from the raw CFA image. $DM$ is the same size as the G plane.
2. Recover color difference planes R-G and B-G using the CFA image and the recovered G plane.
3. Refine the G plane using the initial estimates of color difference planes in step 2 and refine color difference planes using the refined G plane again.
4. Enlarge the edge map $DM$ and enlarge the refined G plane with the aid of the edge map $DM$. This step also estimates part of edge direction information in the new missing G pixel positions, and new estimated edge

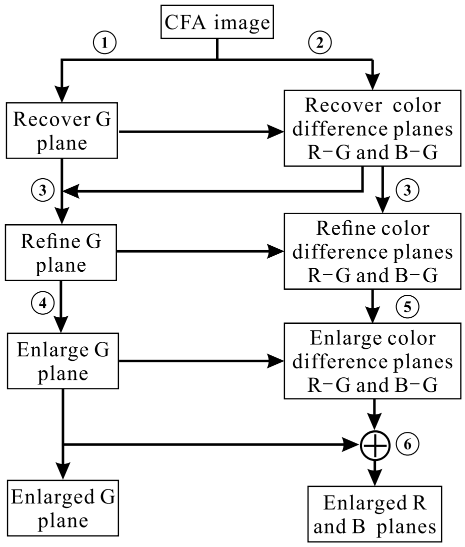

CFA image
1
2
Recover G plane
Recover color difference planes R−G and B−G
3
3
Refine G plane
Refine color difference planes R−G and B−G
4
5
Enlarge G plane
Enlarge color difference planes R−G and B−G
6
Enlarged G plane
Enlarged R and B planes

**Fig. 6** Flow chart of the proposed joint demosaicking and zooming algorithm.

direction information is added to the edge map $DM$ again.

5. Enlarge color difference planes R-G and B-G with the aid of the edge map $DM$.
6. Obtain the enlarged R and B planes by summing the enlarged G plane and enlarged color difference planes R-G and B-G.

It should be noted that the edge map $DM$ and its usage are not included in Fig. 6.

## 3 Experimental Results and Discussions

### 3.1 *Experimental Results*

To evaluate the performance of the proposed algorithm, we used two datasets: the Kodak test image set[8] and the McMaster test image set.[12] The former is the standard test image set, which includes 24 color images of size $512 \times 768$, as shown in Fig. 7. The latter was introduced by Zhang et al.,[11] which includes 18 color images of size $500 \times 500$, as shown in Fig. 8. Although the Kodak test images have been widely used as a benchmark dataset, some works[9–11] have found that the Kodak images have much higher spectral correlation and lower color saturation than typical color images. Contrarily, the McMaster images have lower spectral correlation and more saturated colors. The performance of many state-of-the-art demosaicking algorithms over the Kodak dataset significantly degrades over the McMaster dataset.[11]

The proposed joint demosaicking and zooming algorithm was compared with five recent state-of-the-art algorithms, including the other three joint demosaicking and zooming

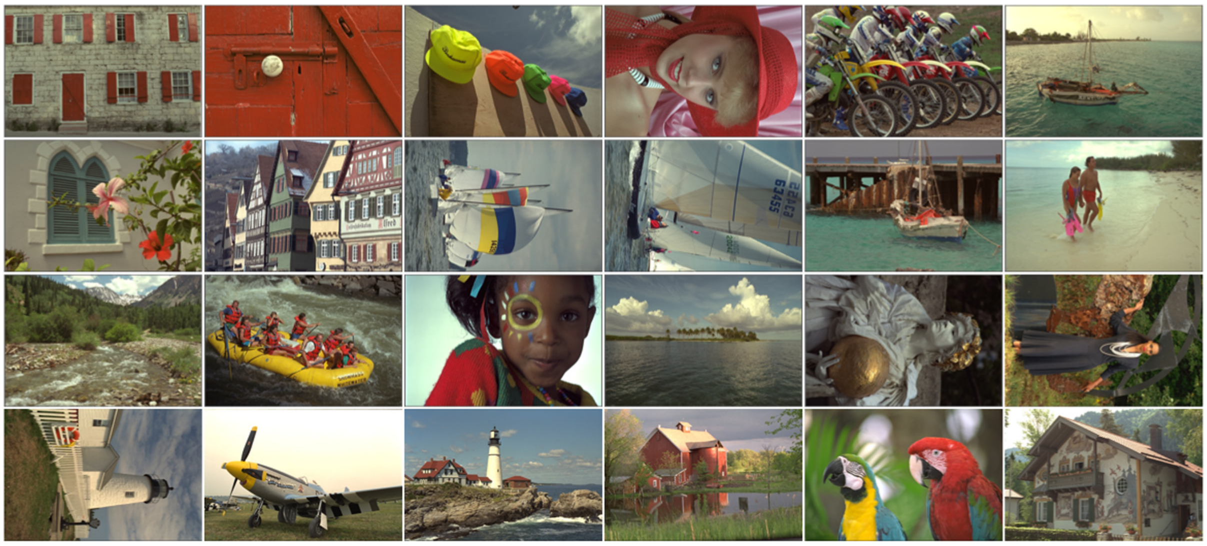


**Fig. 7** Kodak set of test images (refers as images 1 to 24, enumerated from top to bottom and left to right).

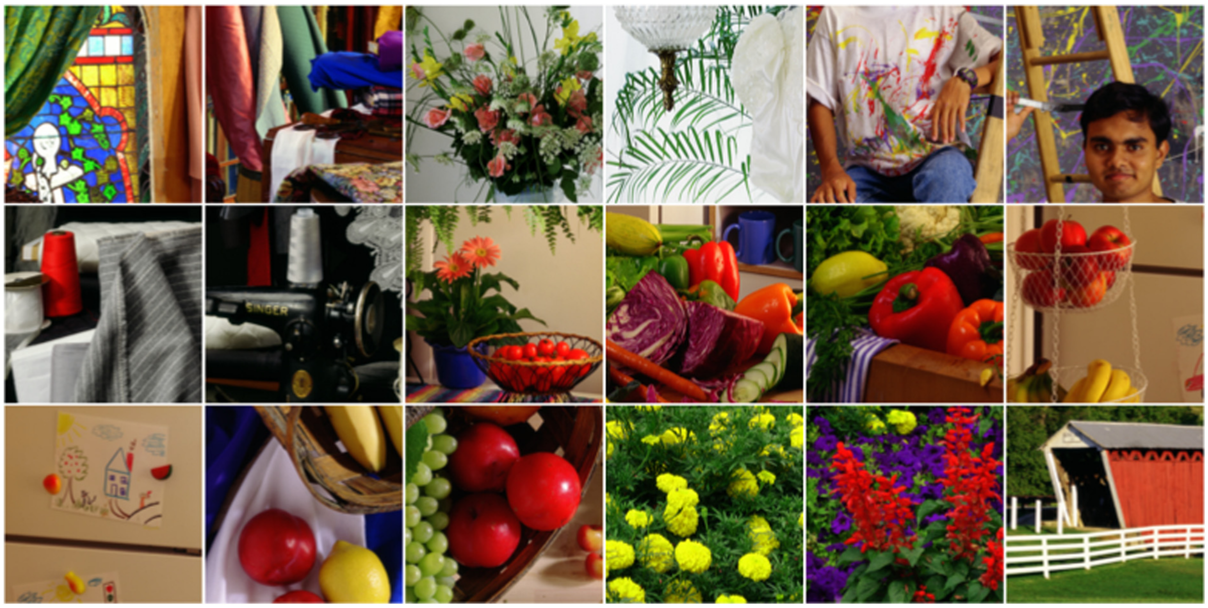


**Fig. 8** McMaster set of test images (refers as images 1 to 18, enumerated from top to bottom and left to right).

algorithms, the first one by Chung and Chan,[5] the second one by Zhang and Zhang,[6] and the last one by Chung et al.[7] They are denoted as A1, A2, and A3, respectively. It has been shown that a joint demosaicking and zooming algorithm is superior to a "demosaicking-first and zooming-later" or "CFA-zooming-first and demosaicking-later" algorithm.[5–7] However, those results are all based on the test images with high spectral correlation. Zhang et al.[11] demonstrated the good demosaicking performance of the LDI-NAT algorithm over the McMaster test images with weak spectral correlation. Here, our results were also compared with those of the "demosaicking-first and zooming-later" scheme based on the LDI-NAT demosaicking.[11] The zooming algorithms used were bilinear and edge-guided image interpolation (EGII).[19] The algorithm in which LDI-NAT is combined with the bilinear interpolation is denoted A4, and the one in which LDI-NAT is combined with EGII interpolation is denoted A5. The source codes or executables of A1, A2,

**Table 1** Color peak signal-to-noise ratio (CPSNR) (in dB) comparison for different algorithms excluding border of 12 pixels from CPSNR computations for McMaster test images. Best CPSNR of each row is shown in bold. The first number in the parentheses of the last row is the rank of the average CPSNR of the algorithm, and the second is the average CPSNR gain of the proposed algorithm in decibels compared with the algorithm.

| Image | A1 | A2 | A3 | A4 | A5 | Proposed |
|---|---|---|---|---|---|---|
| 1 | 21.67 | 21.75 | 21.67 | 22.71 | **22.79** | 22.10 |
| 2 | 27.42 | 27.49 | 27.54 | 27.70 | 27.93 | **27.94** |
| 3 | 23.78 | 24.02 | 23.77 | 23.45 | 23.57 | **24.51** |
| 4 | 25.17 | 25.64 | 25.18 | 25.01 | 25.66 | **26.54** |
| 5 | 25.66 | 25.94 | 25.78 | 27.22 | **27.35** | 26.23 |
| 6 | 27.50 | 27.64 | 27.55 | 29.71 | **29.85** | 28.18 |
| 7 | 28.40 | 28.39 | 28.59 | 27.92 | 27.92 | **28.62** |
| 8 | 28.44 | 28.64 | 28.41 | 28.14 | 28.21 | **28.95** |
| 9 | 27.76 | 28.10 | 28.20 | 28.68 | **28.90** | 28.77 |
| 10 | 30.00 | 30.09 | 29.98 | 31.12 | **31.37** | 31.12 |
| 11 | 31.08 | 31.11 | 31.23 | 32.37 | **32.56** | 32.00 |
| 12 | 29.30 | 29.54 | 29.29 | 29.65 | 29.80 | **30.30** |
| 13 | 32.57 | 32.99 | 32.73 | 33.51 | **33.77** | 33.44 |
| 14 | 32.15 | 32.29 | 32.04 | 32.64 | **32.97** | 32.87 |
| 15 | 32.35 | 32.47 | 32.37 | 33.04 | **33.39** | 33.20 |
| 16 | 24.69 | 24.98 | 24.99 | 25.95 | **25.98** | 25.31 |
| 17 | 23.66 | 23.68 | 24.22 | 25.76 | **25.79** | 24.41 |
| 18 | 26.49 | 26.56 | 26.74 | 26.73 | **27.10** | 27.00 |
| Average | 27.67 | 27.85 | 27.79 | 28.41 | **28.60** | 28.42 |
|  | (6, 0.75) | (4, 0.57) | (5, 0.63) | (3, 0.01) | (1, −0.18) | (2, 0.00) |

**Table 2** CPSNR (in dB) comparison for different algorithms excluding border of 12 pixels from CPSNR computations for Kodak test images. Best CPSNR of each row is shown in bold. The first number in the parentheses of the last row is the rank of the average CPSNR of the algorithm, and the second is the average CPSNR gain of the proposed algorithm in decibels compared with the algorithm.

| Image | A1 | A2 | A3 | A4 | A5 | Proposed |
|---|---|---|---|---|---|---|
| 1 | 24.52 | 24.36 | **24.65** | 24.17 | 24.06 | 24.57 |
| 2 | 30.50 | 30.64 | 30.71 | 30.56 | 30.54 | **30.91** |
| 3 | 31.94 | 31.93 | 32.44 | 31.90 | 32.10 | **32.84** |
| 4 | 30.74 | 30.98 | 31.13 | 30.91 | 30.96 | **31.30** |
| 5 | 24.50 | 24.65 | 24.73 | 24.01 | 24.27 | **25.41** |
| 6 | 25.99 | 25.91 | **26.10** | 25.30 | 25.21 | 26.06 |
| 7 | 30.91 | 31.20 | 31.17 | 30.86 | 31.31 | **32.02** |
| 8 | **21.86** | 21.72 | 21.61 | 21.38 | 21.36 | 21.84 |
| 9 | 30.37 | 30.46 | 30.33 | 30.07 | 30.28 | **30.92** |
| 10 | 30.42 | 30.57 | 30.63 | 30.02 | 30.13 | **31.01** |
| 11 | 27.11 | 27.12 | 27.34 | 26.75 | 26.75 | **27.38** |
| 12 | 31.50 | 31.44 | 31.51 | 31.08 | 31.21 | **31.86** |
| 13 | 22.12 | 22.10 | **22.46** | 21.56 | 21.46 | 22.20 |
| 14 | 25.97 | 26.11 | 26.47 | 26.28 | 26.35 | **26.57** |
| 15 | 30.48 | 30.66 | 30.78 | 30.38 | 30.53 | **31.03** |
| 16 | 29.64 | 29.47 | **29.80** | 28.89 | 28.77 | 29.57 |
| 17 | 30.20 | 30.33 | 30.36 | 29.71 | 29.88 | **30.69** |
| 18 | 26.04 | 26.15 | **26.38** | 25.78 | 25.77 | 26.37 |
| 19 | 26.43 | 26.40 | 26.26 | 25.70 | 25.75 | **26.63** |
| 20 | 29.91 | 29.78 | 29.79 | 29.37 | 29.73 | **30.41** |
| 21 | 26.57 | 26.59 | **26.78** | 26.02 | 25.97 | 26.75 |
| 22 | 27.87 | 28.01 | 27.99 | 27.85 | 27.78 | **28.20** |
| 23 | 32.26 | 32.35 | 32.24 | 31.93 | 32.21 | **33.58** |
| 24 | 24.76 | 24.85 | **24.99** | 24.36 | 24.31 | 24.90 |
| Average | 28.03 | 28.07 | 28.19 | 27.70 | 27.78 | **28.46** |
|  | (4, 0.43) | (3, 0.39) | (2, 0.27) | (6, 0.76) | (5, 0.68) | (1, 0.00) |

A3, LDI-NAT, and EGII can be achieved directly from the original authors. We thank the authors of Refs. 5, 6, 7, 11, and 19 for sharing their codes or programs so that a reliable comparison is possible. A bilinear interpolation algorithm was implemented by MATLAB®'s "INTERP2" function. The parameters of our algorithm used in all the experiments are as follows: $T_1 = 1.5$ in Eq. (2), $c = 2$ in Eqs. (6) and (8), $T_2 = 1.15$ in Eq. (10), and $k = 5$ in Eq. (12). These parameters are determined by training sample images. For more details, please refer to Refs. 13 and 17.

In our experiments, the 24 Kodak test images of size $512 \times 768$ were downsampled to full-color images of size $256 \times 384$ each and then subsampled according to the Bayer CFA pattern to obtain mosaic images. Similarly, the 18 McMaster test images of size $500 \times 500$ were also downsampled to obtain mosaic images of size $250 \times 250$. Subsequently, all mosaic images were processed with different algorithms to produce zoomed full-color images for comparison. Since the original images are known, we can measure the color peak signal-to-noise ratio (CPSNR) of the reconstructed images. CPSNR for a color image with size $M \times N$ is defined by

$$\text{CPSNR} = 10 \log_{10}\left(\frac{255^2}{\text{CMSE}}\right),$$

where

$$\text{CMSE} = \frac{1}{3MN} \sum_{k=1}^{3} \sum_{i=1}^{M} \sum_{j=1}^{N} [I_\text{o}(i,j,k) - I_\text{r}(i,j,k)]^2.$$

$I_\text{o}$ and $I_r$ represent the original and reconstructed images, respectively. Every CPSNR value was calculated, excluding the border of 12 pixels around the image to remove the boundary effects. CPSNR results over the McMaster Images are summarized in Table 1, and CPSNR results over the Kodak Images are summarized in Table 2. The highest CPSNR value in each row is highlighted in bold. In both the tables, the first number in the parentheses of the last row is the rank of the average CPSNR of the algorithm, and the second is the average CPSNR gain of the proposed algorithm in decibels compared with the algorithm.

It can be seen in Table 1 that the proposed algorithm gives the second highest average CPSNR value over the McMaster test images, slightly worse than A5 (0.18 dB). Our algorithm significantly outperforms the other three published joint demosaicking and zooming algorithms A1, A2, and A3. It exceeds the average CPSNR value of the best one among A1, A2, and A3 by 0.57 dB, and the greatest CPSNR improvement is 1.03 dB for image 10. It can also be seen in Table 2 that the proposed algorithm gives the highest average CPSNR value over the Kodak test images and obtains the highest CPSNR values on 16 of the 24 test images. Although A4 and A5 give a good CPSNR performance over the McMaster test images, they have the worst average CPSNR values over the Kodak test images. A5 gives a 0.93 dB better average CPSNR value than A1 over the McMaster test images, but it falls behind A1 by 0.3 dB on the average CPSNR value over the Kodak test images. It is also impressive that the average CPSNR value of A3 is next only to the proposed algorithm over the Kodak test images, but gives the second worst average CPSNR value over the McMaster test images.

It is interesting to compare the LDI-NAT demosaicking results with the intermediate demosaicking results of our algorithm. CPSNR results on the McMaster dataset are shown in Table 3. It can be seen that our average CPSNR falls behind LDI-NAT by 0.3 dB. However, A5, which couples the sophisticated LDI-NAT demosaicking results with the sophisticated EGII interpolation algorithm,[19] only outperforms the proposed joint demosaicking and zooming algorithm by 0.18 dB on the average CPSNR for the McMaster dataset. It can further demonstrate that our joint demosaicking and zooming algorithm is really effective. It should be noted that in our zooming process, zooming the color difference plane only uses the simple directional bilinear interpolation. If the directional high-order interpolation techniques are used, slightly better results can be provided at a higher computational cost.

Figures 9 to 12 show some zooming results. Each includes a cropped region of a test image and the corresponding processing results using different algorithms. Those regions are from the images 12, 3, 18, and 8 in the McMaster dataset. It can be clearly seen that the proposed algorithm achieves the best visual quality and reduces many visible artifacts, including false colors, blurring, aliasing, etc. The proposed algorithm best recovers the edge structures of the cords of the net in Fig. 9 and the peduncles in Fig. 10, and has the fewest false colors. In Fig. 11, A1, A2, and A3 all produce many false spots. Although A4 and A5 almost remove false spots, the resulting images tend to be blurry and have more false colors. Our result has only few false spots [referring to the crossed red wood blocks

**Table 3** CPSNR (in dB) comparison between local directional interpolation and nonlocal adaptive thresholding (LDI-NAT) demosaicking results and the intermediate demosaicking results of our algorithm excluding border of 12 pixels from CPSNR computations for McMaster test images. Better CPSNR in the comparison is shown in bold.

| Image | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LDI-NAT | **23.75** | **29.86** | 25.87 | 27.80 | **27.92** | **30.35** | 31.01 | 31.10 | **30.57** | **32.47** | **33.80** | 32.24 | **35.35** | **34.33** | **34.48** | **27.04** | **25.95** | **30.25** | **30.23** |
| Proposed | 22.70 | 29.76 | **27.85** | **28.66** | 26.09 | 27.89 | **34.38** | **32.72** | 29.77 | 31.79 | 32.82 | **32.43** | 34.19 | 33.94 | 34.02 | 25.84 | 24.00 | 29.95 | 29.93 |

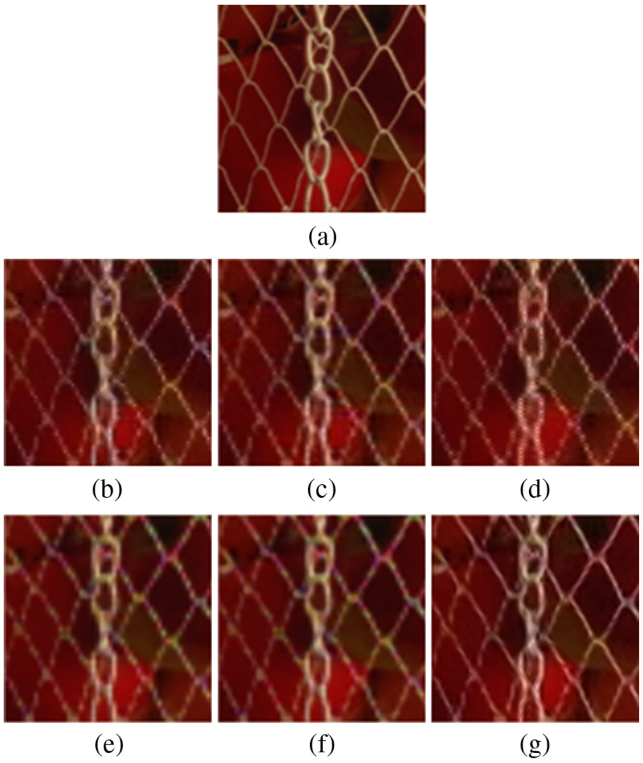


**Fig. 9** Processing results of a cropped region of image 12 in McMaster test images: (a) original, (b) A1, (c) A2, (d) A3, (e) A4, (f) A5, and (g) the proposed algorithm.

on the upside of Fig. 11(g)], has fewer false colors, and better preserves the edges [referring to the white wood strip on the left side of Fig. 11(g)]. In Fig. 12, our result is significantly better than the ones of the other algorithms. All other algorithms produce excessive false colors and serious aliasing artifacts for the knitting. It should be helpful to carefully examine the square regions surrounded by the red lines.

The experimental results are also interesting. LDI-NAT (Ref. 11) is a state-of-the-art demosaicking method on the McMaster dataset. Its average CPSNR is 0.3 dB (Table 3) better than the one of the intermediate demosaicking results of the proposed algorithm. The state-of-the-art edge-guided interpolation method EGII (Ref. 19) following LDI-NAT (i.e., A5) also has an average gain of 0.18 dB (Table 1) over the proposed algorithm on the McMaster dataset. However, the proposed algorithm has better visual quality on the resulting images than A5 (Figs. 9 to 12). The primary reason is that the proposed algorithm uses the consistent edge map in the process of demosaicking and zooming. The edge map can cause the edges of the resulting images to be better preserved and the resulting images to contain fewer artifacts.

### 3.2 *Discussions*

Compared to the algorithms under comparison, the significant advantage of the proposed algorithm is its outstanding performance on both the McMaster and Kodak datasets. Its average CPSNR value is better than those of all others on the Kodak dataset and is next only to that of A5 on the McMaster dataset (0.18 dB). Its average CPSNR value noticeably

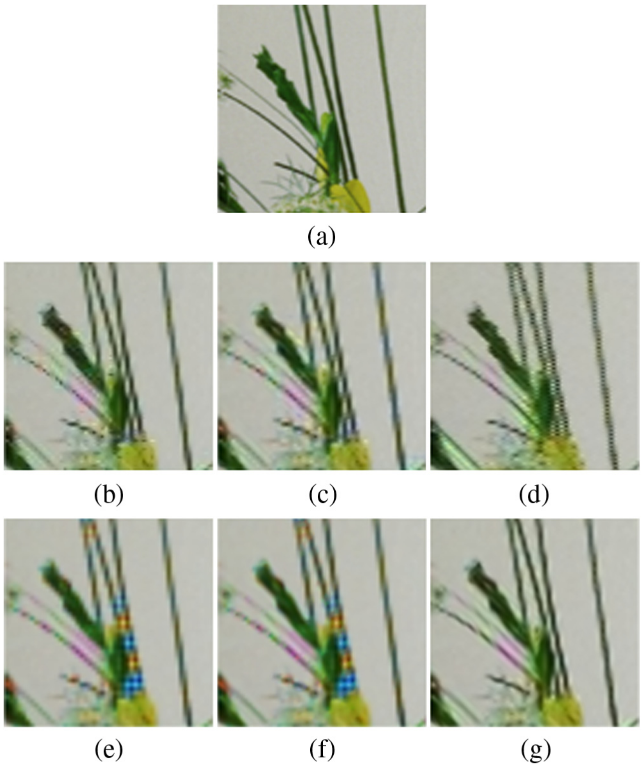


**Fig. 10** Processing results of a cropped region of image 3 in McMaster test images: (a) original, (b) A1, (c) A2, (d) A3, (e) A4, (f) A5, and (g) the proposed algorithm.

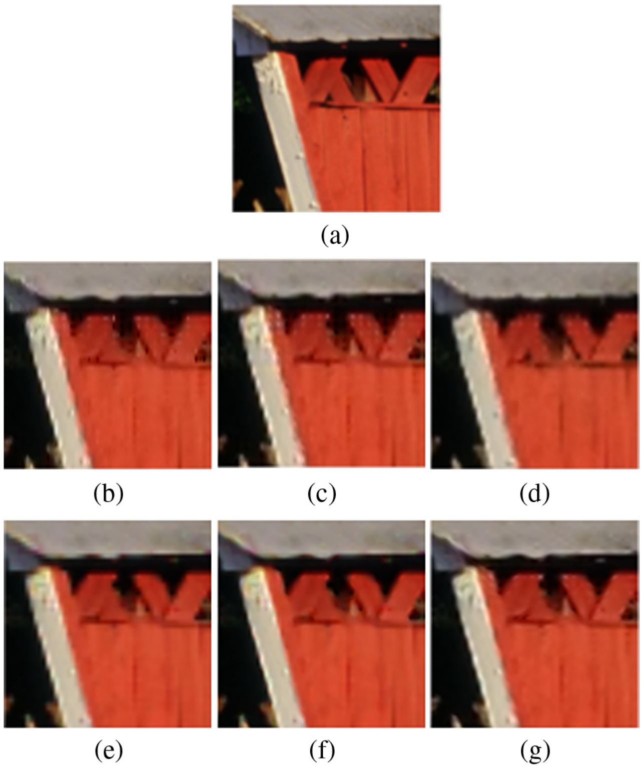
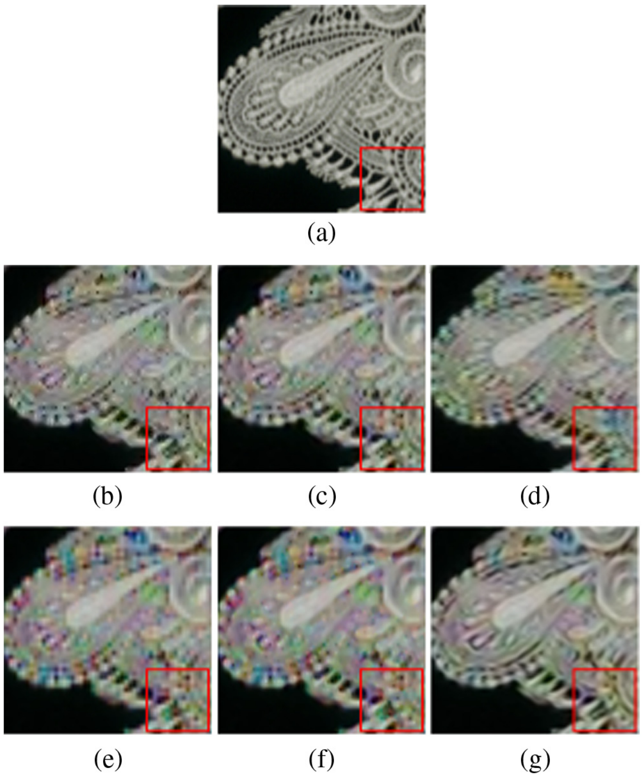


**Fig. 11** Processing results of a cropped region of image 18 in McMaster test images: (a) original, (b) A1, (c) A2, (d) A3, (e) A4, (f) A5, and (g) the proposed algorithm.


(a)

(b) (c) (d)

(e) (f) (g)

**Fig. 12** Processing results of a cropped region of image 8 in McMaster test images: (a) original, (b) A1, (c) A2, (d) A3, (e) A4, (f) A5, and (g) the proposed algorithm.

outperforms that of A5 on the Kodak dataset (0.68 dB), even if A5 is equipped with both the sophisticated LDI-NAT (Ref. 11) demosaicking and the sophisticated EGII (Ref. 19) zooming algorithms. Further, it achieves the highest perceptual quality for the test images. Another advantage of the proposed algorithm is its computational efficiency. To produce a zoomed full-color image from a 250 × 250 CFA image with a zooming factor of 2, we record the running time of different algorithms on a PC with an Intel i5-2300 2.80 GHz CPU (four cores) and 4 GB RAM (only a single core is used). The average running time of our unoptimized MATLAB® implementation is 3.2845 s (the MATLAB® executable (mex) file requires only ∼0.2 s], whereas that of A2 is 27.1565 s. For A5, just the demosaicking time of LDI-NAT takes 167.7958 s, and a following EGII (Ref. 19) zooming requires 34.6737 s. Considering that A1 and A3 are implemented using C/C++, we did not compare with them.

There are two main reasons why our algorithm can work well on both the datasets.

- It “moderately” utilizes the spectral correlation between color planes.
- Both demosaicking and zooming processes share the same edge map.

Unlike the methods in Refs. 5, 6, and 7, which overuse the spectral correlation between color planes, we estimate the edge direction of a missing green sample in the raw CFA data based on the local color difference gradients and estimate color difference planes by the weighted average of the known color difference values in a local neighborhood, and the computations of the weights are also on the color difference planes. All of these alleviate the dependence on the spectral correlation between color planes. Therefore, the performance of our algorithm does not clearly degrade on the McMaster dataset with weak spectral correlation. Contrarily, LDI-NAT cannot fully utilize the strong spectral correlation between color planes of the images in the Kodak dataset; thus, the zooming algorithms dependent on LDI-NAT cannot reach good-enough results on the Kodak dataset.

It should be pointed out that the edge map plays an important role in the performance of the proposed algorithm. With the shared edge map, our algorithm can preserve the same edge directions in the missing pixels for red, green, and blue planes in the demosaicking and zooming processes and, thus, can avoid the additional artifacts on edges. In addition, the directional filtering can be quicky implemented using the edge map; thus, our algorithm also has high computational efficiency.

## 4 Conclusion

Image demosaicking and zooming are two key operations in a digital camera. The previously published algorithms show that demosaicking and zooming, when performed simultaneously instead of separately, can improve the results.[5–7] However, those algorithms all have a serious limitation: they are only adapted to a specific dataset. Generally, they have good performance on the popular Kodak test images, which have high spectral correlation,[10,11] whereas their performance significantly degrades on the McMaster test images,[12] which have weak spectral correlation.[11] On the contrary, the zooming algorithms dependent on the LDI-NAT (Ref. 11) demosaicking have good performance on the McMaster dataset, but their performance seriously deteriorates on the Kodak dataset.

In this paper, a new joint demosaicking and zooming algorithm is proposed for the Bayer CFA pattern, in which the edge direction information (edge map) extracted from the raw CFA data is consistently used in demosaicking and zooming. Its backbone is the reconstruction of the green plane. With the help of the reconstructed green plane and the estimated edge map at the missing green sample positions, the reconstruction of red and blue planes is carried out on color difference planes so as to exploit the spectral correlation between color planes.[1] The demosaicking of the color difference planes is carried out by the weighted average, and the zooming of the color difference planes is carried out by the simple directional bilinear interpolation with the aid of the edge map. Our algorithm shows excellent performance on both the Kodak and McMaster datasets. It also has high computational efficiency. Therefore, it provides a better tradeoff among adaptability, performance, and computational cost compared with the existing algorithms.

### *Acknowledgments*

The authors thank Dr. Y.-H. Chan and Dr. L. Zhang of The Hong Kong Polytechnic University, and Dr. K.-L. Chung of National Taiwan University of Science and Technology for making their MATLAB® codes or executables available online. The authors would also like to thank the anonymous reviewers and the associate editor for their constructive comments and suggestions. This work was supported by the

National Natural Science Foundation of China under Grant Nos. 61372184 and 61172104.

## References

1. B. K. Gunturk et al., "Demosaicking: color filter array interpolation," *IEEE Signal Process. Mag.* **22**(1), 44–54 (2005).
2. D. Menon and G. Calvagno, "Color image demosaicking: an overview," *Signal Process.: Image Commun.* **26**(8–9), 518–533 (2011).
3. R. Lukac, K. N. Plataniotis, and D. Hatzinakos, "Color image zooming on the Bayer pattern," *IEEE Trans. Circuits Syst. Video Technol.* **15** (11), 1475–1492 (2005).
4. R. Lukac, K. Martin, and K. Plataniotis, "Digital camera zooming based on unified CFA image processing steps," *IEEE Trans. Consum. Electron.* **50**(1), 15–24 (2004).
5. K.-H. Chung and Y.-H. Chan, "A low-complexity joint color demosaicking and zooming algorithm for digital camera," *IEEE Trans. Image Process.* **16**(7), 1705–1715 (2007).
6. L. Zhang and D. Zhang, "A joint demosaicking-zooming scheme for single chip digital color cameras," *Comput. Vis. Image Underst.* **107**(1–2), 14–25 (2007).
7. K.-L. Chung et al., "New joint demosaicing and zooming algorithm for color filter array," *IEEE Trans. Consum. Electron.* **55**(3), 1477–1486 (2009).
8. E. Dubois, "24 Kodak Color images," http://www.site.uottawa.ca/~edubois/demosaicking/ (2014).
9. X. Li, B. K. Gunturk, and L. Zhang, "Image demosaicing: a systematic survey," *Proc. SPIE* **6822**, 68221J (2008).
10. F. Zhang et al., "Robust color demosaicking with adaptation to varying spectral correlations," *IEEE Trans. Image Process.* **18**(12), 2706–2717 (2009).
11. L. Zhang et al., "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *J. Electron. Imaging* **20**(2), 023016 (2011).
12. L. Zhang, "18 McMaster Color images," http://www.comp.polyu.edu.hk/~cslzhang/CDM_Dataset.htm (2014).
13. D. Zhou, X. Shen, and W. Dong, "Colour demosaicking with directional filtering and weighting," *IET Image Process.* **6**(8), 1084–1092 (2012).
14. D. Menon, S. Andriani, and G. Calvagno, "Demosaicing with directional filtering and a posteriori decision," *IEEE Trans. Image Process.* **16**(1), 132–141 (2007).
15. J. F. Hamilton and J. E. Adams, "Adaptive color plane interpolation in single sensor color electronic camera," U.S. Patent 5 629 734 (1997).
16. R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Process.* **ASSP-29**(6), 1153–1160 (1981).
17. D. Zhou, X. Shen, and W. Dong, "Image zooming using directional cubic convolution interpolation," *IET Image Process.* **6**(6), 627–634 (2012).
18. Q. Wang and R. K. Ward, "A new orientation-adaptive interpolation method," *IEEE Trans. Image Process.* **16**(4), 889–900 (2007).
19. L. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *IEEE Trans. Image Process.* **15**(8), 2226–2238 (2006).

**Dengwen Zhou** has been a professor in the School of Control and Computer Engineering, North China Electric Power University (NCEPU), China, since October 2002. Prior to NCEPU, he was an associate professor in the Department of Computer Science and Technology, Tsinghua University, China, and a senior engineer with China Aerospace Science and Industry Corporation. His main research interests include image processing, computer vision, and computational photography.

**Weiming Dong** received his BSc and MSc degrees in computer science both from Tsinghua University, China, in 2001 and 2004, respectively, and his PhD degree in computer science from the University of Henri Poincaré Nancy 1, France, in 2007. He is an associate professor in the Sino-French Laboratory and National Laboratory of Pattern Recognition at the Institute of Automation, Chinese Academy of Sciences. His research interests include computer graphics and image analysis.

**Wengang Cheng** received his BS degree from the Shandong University of Science and Technology, China, in 1998, an MS degree from China University of Mining and Technology, Beijing, and a PhD degree in computer science from Beijing Jiaotong University, China, in 2001 and 2005, respectively. He is currently an associate professor in NCEPU, China. His research interests include image and video processing, machine learning, social media computing, etc.