A Low-Complexity Joint Color Demosaicking and Zooming Algorithm for Digital Camera

King-Hong Chung and Yuk-Hee Chan, *Member, IEEE*

***Abstract*—This paper presents a low complexity joint color demosaicking and digital zooming algorithm for single-sensor digital cameras. The proposed algorithm directly extracts edge information from raw sensor data for interpolation in both demosaicking and zooming to preserve edge features in its output. This allows the extracted information to be exploited consistently in both stages and also efficiently, as no separate extraction process is required in different stages. The proposed algorithm can produce a zoomed full-color image as well as a zoomed Bayer color filter array image with outstanding performance as compared with conventional approaches which generally combine separate color demosaicking and digital zooming schemes.**

***Index Terms*—Bayer pattern, color demosaicking, color difference rule, color filter array (CFA), zooming.**

I. Introduction

SINGLE image sensors are widely used nowadays in many portable electronic devices like digital cameras, mobile phones, and personal digital assistants (PDAs) to acquire scene images in digital format. Fig. 1 shows a typical image acquisition system used in these devices. It first allows light passing through an optical system, where focusing, shutter control, and optical zoom are performed, and then acquires the scene with an image sensor under a color filter array (CFA) [1] to produce a digital image in which only one color component is sampled at each pixel location. For reference, an image of such a format is referred to as a CFA image hereafter so as to contrast it from a full-color image in which all three color components are available in each pixel. The Bayer pattern shown in Fig. 2 is the most common CFA used nowadays [2]. To reproduce a full-color image with a CFA image, the two missing color components at each pixel must be estimated by color demosaicking [3]–[10]. Finally, a postprocessing step [11] may be necessary to reduce demosaicking artifacts.

Along with demosaicking, zooming is probably the most commonly performed processing operation in a digital camera. In general, under the size and power constraints, a typical portable device is generally not equipped with a complex optical system to achieve high optical zooming power as such an optical system is expensive and sizable. For this reason, digital zooming is used instead to enhance the zooming power in a digital camera.

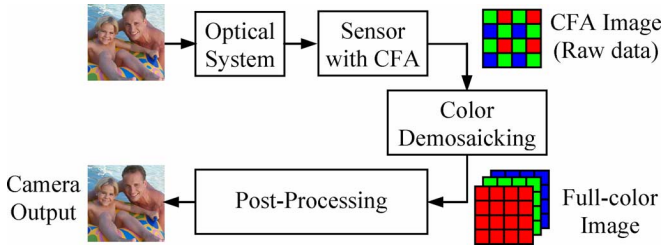


Fig. 1. Image acquisition system in digital camera.

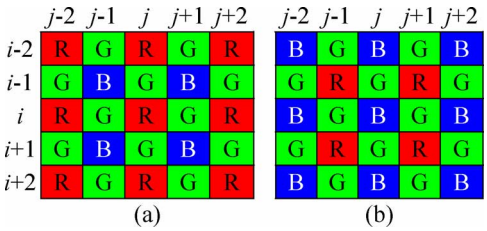


Fig. 2. Two 5 × 5 regions of Bayer pattern having centers at (a) red and (b) blue samples.

Various approaches can be exploited to produce a full-color zooming result from a CFA image. Traditionally, it is done by performing a digital zooming process after demosaicking the CFA image as shown in Fig. 3(a). The zooming process can be performed on the demosaicked full-color image either in a component-wise manner [12]–[15] or in a vector manner [16]–[19]. In a component-wise interpolation, each color plane is treated as a gray-level image and interpolated with a gray-scale image interpolation technique independently. The zoomed full-color image is then formed by combining the three interpolated color planes. As for a vector color interpolation, each pixel in the image forms a color vector with the three color components as elements. Various vector operations are then performed on the color vectors to produce the zoomed full-color image. As the interpolation carried out in the zooming process is based on the demosaicking result and demosaicking may introduce artifacts, such as blurred edges and false colors [11], [20], the quality of the resultant zoomed image can be very poor in regions of complicated details.

Recently, an alternative approach, as shown in Fig. 3(b), was used instead. In this approach, a zooming process is directly applied to the CFA image to generate an enlarged CFA image such

Manuscript received June 3, 2006; revised February 27, 2007. This work was supported in part by the Research Grants Council of Hong Kong Special Administrative Region (PolyU 5205/04E) and in part by the Centre for Signal Processing of The Hong Kong Polytechnic University (POLYU Grant G-U286). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Stanley J. Reeves.

The authors are with the Centre for Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong (e-mail: king.hong.chung@polyu.edu.hk; enyhchan@polyu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2007.898997

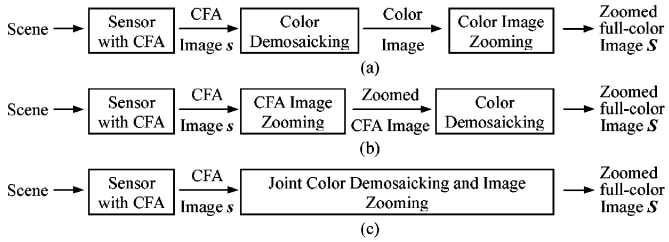


Fig. 3. Image zooming methods: (a) zooming after demosaicking, (b) demosaicking after zooming, and (c) joint demosaicking and zooming.

that a conventional demosaicking technique can be performed on the enlarged CFA image to produce a zoomed full-color image; [21]–[23] shows three of the successful examples. A linear interpolation-based digital zooming method and an adaptive edge-sensing-based digital zooming method are, respectively, introduced in [21] and [22] to operate on the raw sensor data to produce enlarged CFA images for further demosaicking. In [23], local color ratio and edge-sensing weight coefficients are utilized in CFA zooming, demosaicking, and postprocessing to produce the final zoomed full-color image.

The actual process behind both zooming and demosaicking is interpolation and similar signal processing concepts are employed in both cases. When zooming and demosaicking are performed separately, the information available on the raw sensor data is not always utilized consistently and efficiently to yield the enlarged output image. A low-complexity joint color demosaicking and digital zooming algorithm is, hence, proposed in this paper for digital single-sensor cameras to solve this problem, as shown in Fig. 3(c).

By considering that the green channel provides twice the information as compared with the red and the blue channels in a Bayer CFA image [22], the proposed algorithm starts with estimating all missing green samples in the original CFA image. For each missing green sample, it estimates its appropriate interpolation direction based on local intensity gradients and local variances of color-difference values and picks a corresponding linear interpolator to estimate its intensity value. The green plane is then enlarged in a way that all missing green components in the enlarged image are estimated based on 1) the green intensity values of their known or determined neighbors and 2) the interpolation directions of the closest neighbors whose green components are determined in the previous stage. By so doing, the edge information extracted from the raw sensor data for interpolation is used in demosaicking and zooming consistently. It is also used efficiently as no separate extraction process is required in different stages. Finally, the red and the blue missing samples in the enlarged image are estimated with the interpolated green plane and the color difference model used in [10].

Simulation results show that the proposed algorithm is superior to conventional approaches, which are generally combinations of different demosaicking and zooming algorithms, in zooming CFA images and producing zoomed full-color images in terms of output quality at low complexity.

As a remark, we note that the addressed problem can also be considered as an estimation problem in view of that the missing samples in the enlarged full-color image are estimated based on the CFA samples in the course of the joint process. Accordingly, with suitable constraints, standard estimation algorithms, such as a LMMSE estimator, can be used to produce a zoomed full-color image from a CFA image [24]. This approach tries to solve a mathematically defined optimization problem and generally requires some assumptions or training data for deriving the constraints, which is different from the approach on which we focused in this paper.

This paper is structured as follows. In Section II, a color-difference variance-based green plane demosaicking scheme is introduced. In Section III, the details of our joint demosaicking and zooming algorithm is described. In Sections IV and V, simulation results and a complexity analysis of the proposed algorithm are respectively provided. Finally, a conclusion is given in Section VI.

II. Color Difference Variance-Based Green Plane Demosaicking Scheme

The proposed green plane demosaicking scheme processes the pixels one by one in a raster-scanning order. Based on a symmetric local region of the pixel of interest, the scheme determines the interpolation direction and then interpolates the missing green component of the concerned pixel.

In the proposed scheme, the well-known directional second-order Laplacian interpolators proposed by Hamilton and Adams [4], [25] are used to interpolate the missing green component as they were proven to be a simple yet good approximation of the optimal CFA interpolator [7]. In general, in a CFA image, the local region of a pixel without green component is in a pattern as shown in either Fig. 2(a) or (b). Without losing generality, only the former case is considered in this paper. For handling the case shown in Fig. 2(b), one can exchange the roles of red samples and blue samples and then follow the procedures for handling the case shown in Fig. 2(a) to estimate the missing green components.

For the case shown in Fig. 2(a), the missing green component of the center pixel (i,j) is estimated with one of the following directional interpolators:

$$\text{Horizontal}(H): g_{i,j}^{H} = \frac{(G_{i,j-1}+G_{i,j+1})}{2} + \frac{(2R_{i,j}-R_{i,j-2}-R_{i,j+2})}{4} \tag{1}$$

$$\text{Vertical}(V): g_{i,j}^{V} = \frac{(G_{i-1,j}+G_{i+1,j})}{2} + \frac{(2R_{i,j}-R_{i-2,j}-R_{i+2,j})}{4} \tag{2}$$

$$\begin{aligned}\text{Diagonal}(D): g_{i,j}^{D} &= \frac{(G_{i-1,j}+G_{i+1,j}+G_{i,j-1}+G_{i,j+1})}{4} \\ &\quad + \frac{(4R_{i,j}-R_{i-2,j}-R_{i+2,j}-R_{i,j-2}-R_{i,j+2})}{8} \\ &= \frac{g_{i,j}^{H}+g_{i,j}^{V}}{2}\end{aligned} \tag{3}$$

where $R_{m,n}$ and $G_{m,n}$, respectively, denote the known red and green CFA components of pixel (m,n), and $g_{i,j}^{H}$, $g_{i,j}^{V}$, and $g_{i,j}^{D}$

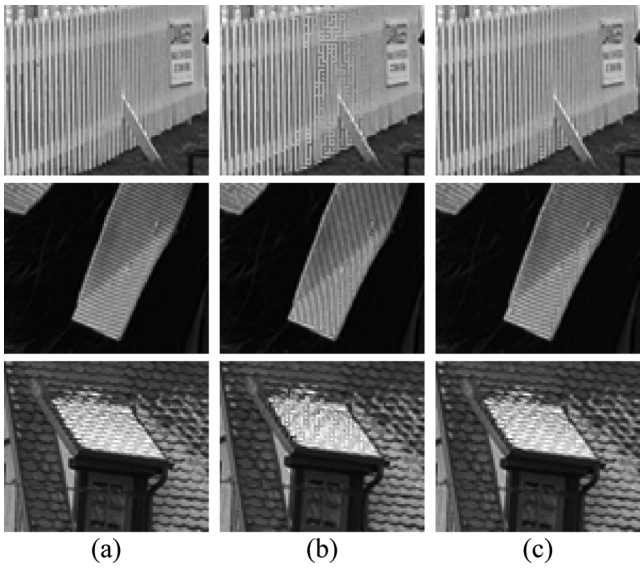


Fig. 4. (a) Original green planes of the original full-color images, (b) green planes generated by algorithm [4], and (c) green planes generated by the proposed green plane demosaicking scheme [the input for generating (b) and (c) were obtained by sampling (a) according to the Bayer CFA pattern].

are the three possible estimates of the missing green component obtained with corresponding interpolators.

The selection of the interpolator is a critical factor to the demosaicking performance. Many adaptive demosaicking methods use local gradients to determine the interpolation direction and, in turn, the interpolator [4], [25], [26]. However, though this determination criterion works very well in simple edge regions, some image pattern features such as those shown in Fig. 4(b) may not be preserved well with this criterion. As the color difference of a pixel (either green-to-red or green-to-blue) usually varies smoothly over a local region in a typical image, the local variance of its value is used as supplementary information to determine the interpolation direction in the proposed scheme when a texture region is encountered.

To reach a better decision, the proposed scheme is a two-pass estimation scheme. In the first pass, the scheme raster-scans the CFA image and detects if a particular pixel is in a sharp horizontal or vertical edge region. If it is, interpolation direction will be determined and the missing green component of the pixel will

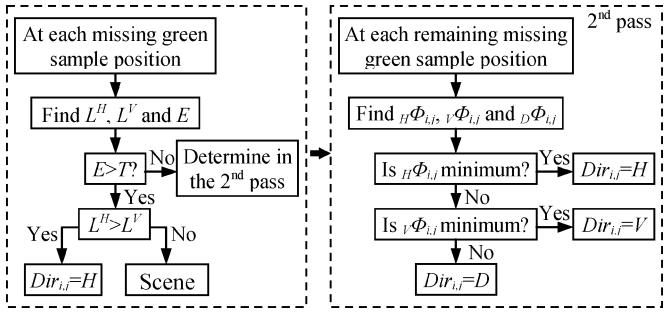


Fig. 5. Procedures for determining the direction to interpolate a missing green component in the proposed green plane demosaicking scheme.

be interpolated accordingly. Otherwise, the pixel will be left behind and processed in the second pass. In the second pass, the color difference information in a local region is used to determine the interpolation direction. Note that, in pass 2, the green components estimated in pass 1 can be used with those known green components to pick appropriate interpolators and perform the interpolation, which helps to improve the interpolation result.

Fig. 5 summarizes how to select an interpolator for a particular pixel in different passes of the proposed scheme. The details are as follows.

A. Pass 1

Both local intensity gradient and color difference are exploited in this pass to evaluate two parameters for a 5×5 local region to see whether there is sharp horizontal or vertical gradient change within the region. For the case shown in Fig. 2(a), the two parameters are computed as (4) and (5), shown at the bottom of the page. The 5×5 window is selected because it matches the support of interpolators (1)–(3). Note that, in both (4) and (5), the first summation term contributes intraband gradient information while the second summation term contributes supplementary interband color information. The second term is used to improve the detection performance in some scenarios when the pixel of interest is on a sharp line of width 1 pixel.

The ratio of L^H and L^V, which is defined to be

$$E = \left| \log_2 \left(\frac{L^V}{L^H} \right) \right| \tag{6}$$

$$L^H = \sum_{n=\pm 2} \left[\sum_{m=0,\pm 2} |R_{i+m,j+n} - R_{i+m,j}| + \sum_{m=\pm 1} |G_{i+m,j+n} - G_{i+m,j}| \right] + \sum_{n=\pm 1} \left[\sum_{m=0,\pm 2} |G_{i+m,j+n} - R_{i+m,j}| + \sum_{m=\pm 1} |B_{i+m,j+n} - G_{i+m,j}| \right] \tag{4}$$

$$L^V = \sum_{m=\pm 2} \left[\sum_{n=0,\pm 2} |R_{i+m,j+n} - R_{i,j+n}| + \sum_{n=\pm 1} |G_{i+m,j+n} - G_{i,j+n}| \right] + \sum_{m=\pm 1} \left[\sum_{n=0,\pm 2} |G_{i+m,j+n} - R_{i,j+n}| + \sum_{n=\pm 1} |B_{i+m,j+n} - G_{i,j+n}| \right] \tag{5}$$

is then used to classify the region and determine the dominant edge direction. A region is said to be a sharp edge region if its associated E is larger than a predefined threshold value T. Our experimental results showed that $T = 1$ provided a good detection result. We note that (6) is not a completely symmetric function with respect to L^V/L^H. However, this small asymmetry does not affect its detection performance.

If the pixel of interest, say pixel (i, j), locates at the center of a sharp edge region, its interpolation direction $Dir_{i,j} \in \{H, V, D\}$, as well as its green estimate $g_{i,j}$, are determined as follows:

$$\begin{cases} Dir_{i,j} = H \text{ and } g_{i,j} = g^H_{i,j}, & \text{if } 2L^H < L^V \\ Dir_{i,j} = V \text{ and } g_{i,j} = g^V_{i,j}, & \text{if } L^H > 2L^V \end{cases}. \quad (7)$$

Otherwise, it is left behind for being processed in Pass 2.

As will be discussed later, to help selecting an interpolator for a pixel, sometimes it is necessary for one to make a preliminary estimation of some other pixels' missing green components. Unlike these estimates which are temporarily obtained for selecting an interpolator, the estimate obtained in (7) is determined with the selected interpolator. Accordingly, it is referred to as a formal estimate for future reference.

Once a missing green component is handled, the same process is performed for estimating the next missing green component in a raster scan manner. For estimating the missing green component in the case shown in Fig. 2(b), one can replace the red samples by the corresponding blue samples and follow the procedures above to determine its interpolation direction and its interpolated value. At the end of the first pass, the missing green components of the center pixels in all 5×5 sharp edge regions should be determined.

B. Pass 2

In the second pass, all missing green components which have not yet been estimated in the first pass are processed in a raster scan manner. The involved pixels are considered to be in a flat region or a texture region (nonedge region). As a region of size 5×5 does not provide enough information to determine the interpolation direction in pass 1, the local region of interest is extended to 9×9 pixels in pass 2 to cover more samples such that more useful information can be extracted at a reasonable increase in computation cost. In fact, based on our empirical study, we found that a region of 9×9 pixels provided the best overall zooming performance in terms of signal-to-noise ratio as compared with other possible sizes such as 13×13.

For the case shown in Fig. 2(a), to determine the interpolation direction for the center pixel, the variances of the color difference values of the pixels along the axes of the extended 9×9 local region, say ${}_H\Phi_{i,j}$ and ${}_V\Phi_{i,j}$, are computed. Fig. 6 shows the pixels involved in the computation. In formulation, ${}_H\Phi_{i,j}$ and ${}_V\Phi_{i,j}$ are defined as

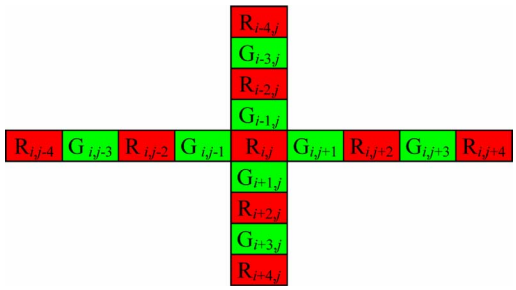


Fig. 6. Pixels along the horizontal and vertical axes of a 9×9 window in a Bayer CFA image.

$$ {}_H\Phi_{i,j} = \sum_{m=-2}^{2} \left(d_{i,j+2m} - \frac{1}{5} \sum_{n=-2}^{2} d_{i,j+2n} \right)^2 $$
$$ \text{and } {}_V\Phi_{i,j} = \sum_{m=-2}^{2} \left(d_{i+2m,j} - \frac{1}{5} \sum_{n=-2}^{2} d_{i+2n,j} \right)^2 \quad (8)$$

where $d_{p,q}$ is a color difference value of pixel (p, q). The values of $d_{i,j+2n}$ and $d_{i+2n,j}$ for $n \in \{0, \pm 1, \pm 2\}$ are computed as follows in (9) and (10), shown at the bottom of the page. Note that the green components for computing $d_{p,q} \in \{d_{i,j+2n}, d_{i+2n,j} | n = -1, -2\}$ have been estimated before as all missing green components are estimated in a raster scan order. As for those for computing $d_{p,q} \in \{d_{i,j+2n}, d_{i+2n,j} | n = 1, 2\}$, they may have been estimated in the first pass. If they are not, the preliminary estimates $g^H_{i,j+2n}$ and $g^V_{i+2n,j}$, which are obtained with (1) and (2), will be, respectively, used to compute $d_{i,j+2n}$ and $d_{i+2n,j}$.

Sometimes, neither a horizontal nor a vertical interpolator can provide a good estimation result and the diagonal interpolator defined in (3) is preferred. Another color difference variance of the pixels within the 9×9 window, say ${}_D\Phi_{i,j}$, is used to detect this situation. In formulation, it is defined as

$$ {}_D\Phi_{i,j} = \frac{1}{2} \left(\sum_{m=-2}^{2} \left(d_{i,j+2m} - \frac{1}{5} \sum_{n=-2}^{2} d_{i,j+2n} \right)^2 + \sum_{m=-2}^{2} \left(d_{i+2m,j} - \frac{1}{5} \sum_{n=-2}^{2} d_{i+2n,j} \right)^2 \right). \quad (11)$$

$$ d_{i,j+2n} = \begin{cases} R_{i,j+2n} - g_{i,j+2n}, & \text{if } g_{i,j+2n} \text{ has already been estimated} \\ R_{i,j+2n} - g^H_{i,j+2n}, & \text{otherwise} \end{cases} \quad (9)$$

$$ d_{i+2n,j} = \begin{cases} R_{i+2n,j} - g_{i+2n,j}, & \text{if } g_{i+2n,j} \text{ has already been estimated} \\ R_{i+2n,j} - g^V_{i+2n,j}, & \text{otherwise} \end{cases} \quad (10)$$

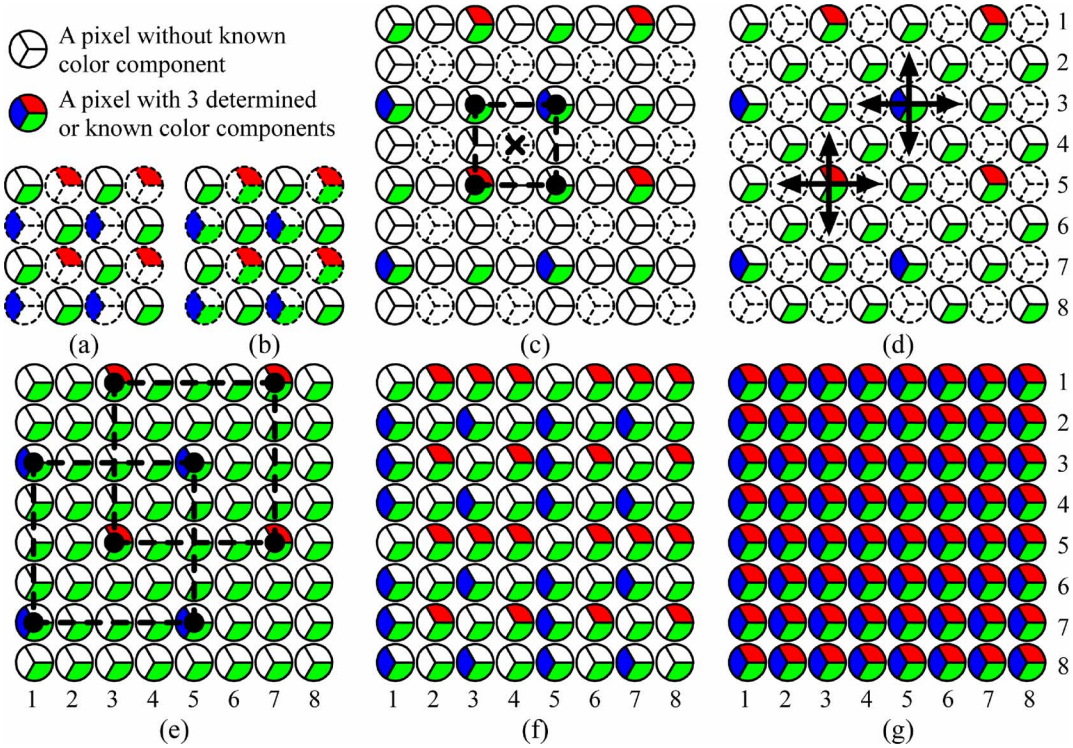


Fig. 7. Spatial arrangement of the intermediate results of the proposed joint demosaicking and zooming algorithm: (a) raw sensor output CFA image, (b) after demosaicking green-plane, (c) after spatial expansion, (d) after estimating diagonal green components, (e) after estimating all green components, (f) intermediate result containing the enlarged CFA image, and (g) final enlarged full-color image.

Similar to the computation of ${}_H\Phi_{i,j}$ and ${}_V\Phi_{i,j}$, to get $d_{p,q} \in \{d_{i,j+2n}, d_{i+2n,j} | n = 0, \pm 1, \pm 2\}$ for evaluating ${}_D\Phi_{i,j}$, the formal estimates of the involved missing green components will be used if they exist. Otherwise, the preliminary estimate $g^D_{p,q}$ obtained with (3) is used to get $d_{p,q} = R_{p,q} - g^D_{p,q}$.

With parameters ${}_H\Phi_{i,j}$, ${}_V\Phi_{i,j}$ and ${}_D\Phi_{i,j}$, the desired interpolator for estimating the missing green component in the middle of the region is selected and its formal estimate is determined by (12), shown at the bottom of the page.

For the case shown in Fig. 2(b), one can treat blue samples as red samples and follow the above procedures to estimate the missing green component. A complete demosaicked green plane is obtained after Pass 2. Fig. 4(c) shows some green planes obtained with the proposed green plane demosaicking scheme.

III. Joint Demosaicking and Zooming Algorithm

As the green channel provides twice information as compared with the red and the blue channels in a CFA image, the proposed joint demosaicking and zooming algorithm starts with green-plane interpolation. The red and the blue plane interpolations then follow with reference to the interpolated green plane. Assume that a CFA image $\boldsymbol{s}$ of size $M \times N$ has to be enlarged to a zoomed full-color image $\boldsymbol{S}$ of size $\lambda M \times \lambda N$. The proposed algorithm supports a zooming factor $\lambda = 2^k$, where k is a positive integer. In this paper, $\lambda = 2$ is selected for simplicity to facilitate the following discussion.

For the sake of reference, hereafter, a pixel at location (m, n) in image $\boldsymbol{s}$ is denoted by $s_{m,n} = \{s_{r(m,n)}, s_{g(m,n)}, s_{b(m,n)}\}$ with $s_{r(m,n)}$, $s_{g(m,n)}$ and $s_{b(m,n)}$ as its red, green and blue components, while a pixel at location (m, n) in image $\boldsymbol{S}$ is represented by $S_{m,n} = \{S_{r(m,n)}, S_{g(m,n)}, S_{b(m,n)}\}$ with $S_{r(m,n)}$, $S_{g(m,n)}$ and $S_{b(m,n)}$ as the corresponding color components.

Fig. 7 shows how image $\boldsymbol{S}$ is obtained using CFA image $\boldsymbol{s}$ step by step in different stages for reference. In particular, the circles with segmented edges denote the pixels to be processed in the following processing stage, and the existence of the color in a division denotes that the corresponding color component is known or has been estimated in a previous processing stage.

A. Green (G) Plane Interpolation

In the proposed algorithm, the green plane demosaicking scheme described in Section II is carried out first to estimate the missing green components in the small CFA image $\boldsymbol{s}$. It results in a direction map indicating the interpolation directions

$$\begin{cases} Dir_{i,j} = H \text{ and } g_{i,j} = g^H_{i,j} \text{ if } {}_H\Phi_{i,j} = \min({}_H\Phi_{i,j}, {}_V\Phi_{i,j}, {}_D\Phi_{i,j}) \\ Dir_{i,j} = V \text{ and } g_{i,j} = g^V_{i,j} \text{ if } {}_V\Phi_{i,j} = \min({}_H\Phi_{i,j}, {}_V\Phi_{i,j}, {}_D\Phi_{i,j}) \\ Dir_{i,j} = D \text{ and } g_{i,j} = g^D_{i,j} \text{ if } {}_D\Phi_{i,j} = \min({}_H\Phi_{i,j}, {}_V\Phi_{i,j}, {}_D\Phi_{i,j}) \end{cases} \tag{12}$$

for the missing green components and a complete demosaicked green plane of image $\boldsymbol{s}$ as shown in Fig. 7(b).

Median filtering is then applied to the color difference planes of image $\boldsymbol{s}$ to refine the demosaicked green plane. For a pixel which is in the middle of the pattern shown in Fig. 2(a), its demosaicked green component $s_{g(i,j)}$ is refined by

$$s_{g(i,j)} = \text{median}(d_{(g-r)(i,j)}, d_{(g-r)(i-2,j)}, d_{(g-r)(i+2,j)}, d_{(g-r)(i,j-2)}, d_{(g-r)(i,j+2)}) + s_{r(i,j)} \quad (13)$$

where $d_{(g-r)(m,n)} = s_{g(m,n)} - s_{r(m,n)}$ is the green-to-red color difference of the pixel at location (m,n) and $\text{median}(\bullet)$ denotes the median operator which provides the median value of its given inputs. Note that all $s_{r(m,n)}$ involved in (13) are raw sensor components in image $\boldsymbol{s}$. Median filtering is used here to enhance the demosaicked green plane because of its simplicity and its proven good performance in handling this case. To refine a pixel which is in the middle of the pattern shown in Fig. 2(b), the same refinement step can be used after replacing red samples with corresponding blue samples.

The enhanced partially demosaicked image $\boldsymbol{s}$ is then expanded to form an image of the same size as the zoomed image $\boldsymbol{S}$. In particular, we have

$$S_{2i-1,2j-1} = s_{i,j} \qquad ,\forall (i,j) \text{ in image } \boldsymbol{s}. \quad (14)$$

Fig. 7(c) shows the spatial arrangement of the output of this stage.

In Fig. 7(c), three quarters of the green components are missing. Those of the pixels marked with segmented edges are interpolated first as each one of them is surrounded by four known green components which form a square support region as shown in Fig. 7(c). Let $S_{g(p,q)}$ be a green component to be interpolated in this stage and $\zeta_{p,q} = \{(p-1,q-1),(p-1,q+1),(p+1,q-1),(p+1,q+1)\}$ be its support. Its value is determined by

$$S_{g(p,q)} = \frac{\left(\sum_{(a,b)\in\zeta_{p,q}} w_{a,b} S_{g(a,b)}\right)}{\left(\sum_{(a,b)\in\zeta_{p,q}} w_{a,b}\right)} \quad (15)$$

where $w_{a,b}$ is a weighting factor defined as

$$w_{a,b} = \sum_{\substack{(m,n)\in\zeta_{p,q} \\ (m,n)\neq(a,b)}} \left(D_{\max} - \delta_{a,b}^{m,n}\right) + 1. \quad (16)$$

In (16), $\delta_{a,b}^{m,n} = \left|S_{g(m,n)} - S_{g(a,b)}\right|$ is actually the absolute difference between two known green components in $\zeta_{p,q}$ and $D_{\max}$ is the maximum value in $\{\delta_{i,j}^{u,v} | (u,v) \neq (i,j) \text{ and } (u,v),(i,j) \in \zeta_{p,q}\}$. Note that the weights are determined in a way that a known green component $S_{g(a,b)}$ will contribute less to the estimate of $S_{g(p,q)}$ if it is dissimilar to the other known green components in $\zeta_{p,q}$. This idea is similar to that of the edge-sensing weighting scheme used in [22], [23] where the weights are computed as the reciprocal values of gradients. However, the complexity of the proposed scheme is lower as division is not required to determine the weights. Fig. 7(d) shows the spatial arrangement of the output of this stage.

Fig. 8. Original full-color images (refers as image 1 to image 24, from top to bottom and left to right).

The remaining missing green components in Fig. 7(d) are then interpolated by simple directional bilinear interpolation. Let the location of the pixel whose green component is currently being estimated be (p,q). As shown in Fig. 7(d), for each pixel having a missing green component at this stage, there is a neighboring pixel whose green component was determined when demosaicking the green plane of CFA image $\boldsymbol{s}$. Let the location of this neighboring pixel of $S_{p,q}$ be (i,j). $S_{g(p,q)}$ is then estimated as

$$S_{g(p,q)} = \begin{cases} \frac{(S_{g(p,q-1)} + S_{g(p,q+1)})}{2}, & \text{if } \Gamma(S_{g(i,j)}) = H \\ \frac{(S_{g(p-1,q)} + S_{g(p+1,q)})}{2}, & \text{if } \Gamma(S_{g(i,j)}) = V \\ \frac{(S_{g(p,q-1)} + S_{g(p,q+1)} + S_{g(p-1,q)} + S_{g(p+1,q)})}{4}, & \text{if } \Gamma(S_{g(i,j)}) = D \end{cases} \quad (17)$$

where $\Gamma(S_{g(i,j)})$ denotes the interpolation direction for estimating $S_{g(i,j)}$ when demosaicking the green plane of $\boldsymbol{s}$.

In (17), by taking the advantage of the edge consistency in a small region, missing green components in a local region are interpolated with the same direction. This process provides a simple but effective means to preserve edge feature in the zooming result. Fig. 7(e) shows the spatial arrangement of the output of this stage where all the missing green components of image $\boldsymbol{S}$ are determined.

B. Red (R) Plane and Blue (B) Plane Interpolation

To constitute the missing red and blue components in Fig. 7(e), the color difference model used in [10] is employed. For each pixel whose red (blue) component is missing, its green-to-red (green-to-blue) color difference value is bilinearly interpolated from the neighboring pixels having known red (blue) CFA components and its intensity value can then be determined.

For example, when the red components of pixels (p,q), $(p,q+4)$, $(p+4,q)$, and $(p+4,q+4)$ are known and one

TABLE I
CPSNR PERFORMANCE (dB) OF VARIOUS ALGORITHMS IN PRODUCING A ZOOMED FULL-COLOR IMAGE

Image	ACPI[4] + BI	AP[5] + BI	PCSD[6] + BI	AHDA[7] + BI	NCED[8] + BI	BICD[9] + BI	DCZPS [23]	CIZBP[22] + NCED[8]	CIZBP[22] + AP[5]	CIZBP[22] +AHDA[7]	Ours
1	23.74	24.44	23.97	24.20	24.40	21.60	22.52	22.55	22.23	22.69	**24.49**
2	30.24	30.46	30.36	30.42	**30.69**	28.61	29.06	29.12	28.88	29.24	30.51
3	31.20	31.67	31.78	31.87	**32.18**	29.67	30.17	30.25	29.86	30.37	32.02
4	30.31	30.84	30.72	30.65	**31.18**	28.49	28.89	28.82	28.55	28.95	30.79
5	23.28	24.33	24.16	24.06	**24.50**	21.29	22.05	21.91	21.60	22.08	24.46
6	24.79	25.63	25.71	25.73	25.58	23.37	24.06	23.93	23.67	24.09	**25.97**
7	30.33	30.76	30.68	30.72	**31.17**	27.23	28.07	28.11	27.71	28.22	30.90
8	21.01	21.57	21.39	21.53	21.56	18.84	19.76	19.81	19.50	19.95	**21.82**
9	29.69	30.14	30.15	30.22	30.30	26.79	27.87	28.07	27.65	28.25	**30.51**
10	29.49	30.54	30.26	30.10	30.54	27.38	28.18	28.08	27.78	28.23	**30.61**
11	26.27	26.97	26.79	26.84	27.04	24.50	25.24	25.19	24.90	25.32	**27.07**
12	30.67	31.20	31.19	31.24	31.31	28.64	29.37	29.39	29.05	29.57	**31.49**
13	21.03	22.02	21.71	21.70	22.08	19.82	20.37	20.12	19.91	20.25	**22.10**
14	25.72	25.95	25.81	25.93	**26.40**	24.09	24.59	24.52	24.23	24.63	25.95
15	29.91	30.44	30.30	30.29	**30.67**	28.34	28.66	28.67	28.31	28.79	30.57
16	28.39	29.21	29.33	29.31	29.19	27.04	27.63	27.58	27.31	27.72	**29.61**
17	29.23	30.14	29.86	29.87	**30.24**	27.17	27.89	27.81	27.51	27.96	30.23
18	25.36	26.22	25.86	25.86	**26.35**	23.69	24.36	24.17	23.92	24.30	26.12
19	25.50	26.03	26.19	26.21	25.87	22.95	23.99	24.19	23.85	24.33	**26.49**
20	28.83	29.43	29.22	29.33	29.65	26.62	27.51	27.66	27.24	27.85	**29.89**
21	25.51	26.37	26.16	26.18	26.45	23.78	24.53	24.39	24.09	24.53	**26.55**
22	27.29	27.90	27.64	27.66	**28.04**	25.63	26.27	26.16	25.91	26.29	27.81
23	31.30	31.79	31.98	32.10	32.16	28.94	29.48	29.44	29.08	29.58	**32.20**
24	23.79	24.78	24.50	24.43	**24.81**	22.50	23.04	22.77	22.56	22.92	24.80
Avg.	**27.20**	**27.87**	**27.74**	**27.77**	**28.02**	**25.29**	**25.98**	**25.95**	**25.64**	**26.09**	**28.04**

wants to estimate the red component of pixel $(p+m, q+n)$ for $0 \le m, n \le 4$, the green-to-red color difference value of pixel $(p+m, q+n)$ is first interpolated as follows:

$$\begin{aligned} & d_{(g-r)(p+m,q+n)} \\ &= \frac{4-n}{4}\left(\frac{m}{4}d_{(g-r)(p+4,q)} + \frac{4-m}{4}d_{(g-r)(p,q)}\right) \\ &\quad + \frac{n}{4}\left(\frac{m}{4}d_{(g-r)(p+4,q+4)} + \frac{4-m}{4}d_{(g-r)(p,q+4)}\right) \\ &\quad \text{for } 0 \le m, n \le 4. \end{aligned} \tag{18}$$

The missing red color component is then estimated by

$$S_{r(p+m,q+n)} = S_{g(p+m,q+n)} - d_{(g-r)(p+m,q+n)}. \tag{19}$$

In practice, the interpolation of all missing red and blue components can be done in parallel so as to reduce the processing time. However, when the components are processed sequentially, one can interpolate some of them first to obtain an intermediate result as shown in Fig. 7(f). This intermediate result contains all color components of image $\boldsymbol{S}$ in CFA format, which allows image $\boldsymbol{S}$ to be stored in CFA format well before its full-color image is finally determined as shown in Fig. 7(g).

Unlike the algorithm proposed in [23] where the color difference information used in the CFA zooming process is obtained from color components located at different positions, the proposed algorithm extract color difference information from color components of the same pixel. This helps to eliminate color artifacts in the resultant image.

IV. SIMULATION RESULTS

Simulation was carried out to evaluate the performance of the proposed joint demosaicking and zooming algorithm. Some other methods for generating zoomed full-color images were also evaluated for comparison. Among them, Lukac's methods (CIZBP [22] and DCZPS [23]) were examples of the approach shown in Fig. 3(b). For the approach shown in Fig. 3(a), various demosaicking algorithms such as ACPI [4], AP [5], PCSD [6], AHDA [7], NCED [8], and BICD [9] were combined with bilinear image zooming methods (BI) [13] to produce zoomed full-color images.

Note that some other interpolation algorithms such as cubic interpolation can also work with NCED, PCSD, AHDA, AP, ACPI, and BICD to provide a zoomed full-color image. However, BI was used in our simulations as low complexity is one of the key concerns in our study. Other interpolation algorithms are comparatively more complicated. In fact, after demosaicking, sophisticated edge-sensing interpolation algorithms may not provide better zooming performance as compared with BI. When demosaicking and zooming are separately performed, the artifacts introduced during demosaicking may fool an edge-sensing interpolator and an interpolation in a wrong direction may even amplify the artifacts.

Fig. 8 shows 24 original 24-bit (8-bit for each color component) full-color images used in the simulation. Each of them is of size 512×768 pixels. They were down-sampled by pixel omission to full-color images of size 256×384 each and then subsampled according to the Bayer CFA pattern, with starting sampling sequence of "GRGR…" in the first row, to form a set of small CFA testing images. The CFA testing images were then processed with different evaluated methods to produce zoomed full-color images for comparison.

TABLE II
CIELAB COLOR DIFFERENCE PERFORMANCE OF VARIOUS ALGORITHMS IN PRODUCING A ZOOMED FULL-COLOR IMAGE

Image	ACPI[4] + BI	AP[5] + BI	PCSD[6] +BI	AHDA[7] + BI	NCED[8] + BI	BICD[9] + BI	DCZPS [23]	CIZBP[22] + NCED[8]	CIZBP[22] + AP[5]	CIZBP[22] +AHDA[7]	Ours
1	7.802	6.540	7.083	6.073	6.374	11.075	8.646	8.269	8.851	8.158	**5.985**
2	3.500	3.460	3.436	3.250	**3.211**	4.342	3.812	3.670	3.842	3.660	3.267
3	2.786	2.555	2.477	**2.288**	2.309	3.430	3.022	2.833	3.047	2.836	2.363
4	3.736	3.411	3.478	3.282	**3.117**	4.543	4.106	3.970	4.188	3.974	3.294
5	8.637	7.189	7.347	6.669	**6.546**	11.453	9.265	8.797	9.580	8.751	6.749
6	6.097	4.993	4.796	**4.361**	4.901	7.927	6.168	6.086	6.481	5.988	4.373
7	3.190	3.166	3.059	2.943	**2.756**	4.736	4.400	4.066	4.379	4.136	2.944
8	8.904	7.796	7.799	7.029	7.536	13.430	10.646	9.926	10.763	9.832	**6.776**
9	3.239	2.964	2.869	2.689	2.770	4.556	3.873	3.577	3.835	3.560	**2.679**
10	3.278	2.829	2.870	2.702	2.716	4.489	3.746	3.490	3.754	3.465	**2.689**
11	5.175	4.470	4.502	**4.102**	4.237	6.932	5.536	5.302	5.706	5.264	4.116
12	2.962	2.637	2.598	**2.406**	2.500	3.826	3.224	3.062	3.271	3.033	2.415
13	10.932	8.566	9.173	8.057	8.227	13.568	10.514	10.605	11.206	10.395	**7.976**
14	6.268	5.668	5.765	5.266	**5.169**	8.064	6.740	6.499	6.951	6.458	5.335
15	3.529	3.185	3.236	3.049	**2.992**	4.234	3.831	3.679	3.919	3.692	3.070
16	4.217	3.506	3.326	3.059	3.437	5.432	4.348	4.248	4.526	4.196	**3.058**
17	3.595	2.998	3.140	2.851	**2.832**	4.865	4.126	3.909	4.216	3.874	2.863
18	6.403	5.471	5.779	5.312	**5.127**	8.123	6.637	6.579	6.952	6.499	5.395
19	5.057	4.400	4.325	3.954	4.321	7.451	5.858	5.553	5.920	5.511	**3.920**
20	3.293	2.920	2.992	2.724	2.700	4.324	3.601	3.361	3.668	3.369	**2.696**
21	5.597	4.655	4.826	4.329	4.408	7.266	5.814	5.665	6.076	5.602	**4.307**
22	5.106	4.614	4.735	4.480	**4.343**	6.289	5.313	5.236	5.494	5.200	4.522
23	2.618	2.568	2.517	2.382	**2.366**	3.171	3.051	2.938	3.085	2.940	2.432
24	6.321	5.308	5.516	5.020	4.978	8.172	6.556	6.376	6.820	6.295	**4.953**
Avg.	**5.093**	**4.411**	**4.485**	**4.095**	**4.161**	**6.737**	**5.535**	**5.321**	**5.689**	**5.279**	**4.091**

TABLE III
AVERAGE PERFORMANCE OF VARIOUS COMBINATIONS IN PRODUCING A ZOOMED FULL-COLOR IMAGE WHEN THE ZOOMING-AFTER-DEMOSAICKING APPROACH IS USED

Zooming algorithm	Demosaicking algorithm					
	ACPI [4]	**AP [5]**	**PCSD [6]**	**AHDA [7]**	**NCED [8]**	**BICD [9]**
Bilinear (BI)	27.20	27.87	27.74	27.77	28.02	25.29
Cubic [28]	27.12	27.86	27.68	27.70	28.03	25.26
Edge-apdative [29]	26.72	27.48	27.29	27.46	27.61	24.69

(a) CPSNR (dB)

Zooming algorithm	Demosaicking algorithm					
	ACPI [4]	**AP [5]**	**PCSD [6]**	**AHDA [7]**	**NCED [8]**	**BICD [9]**
Bilinear (BI)	5.093	4.411	4.485	4.095	4.161	6.737
Cubic [28]	5.230	4.432	4.564	4.125	4.174	7.017
Edge-apdative [29]	5.429	4.538	4.737	4.212	4.312	7.434

(b) CIELab color difference

Table I tabulates the performance of various methods in terms of the color-peak signal-to-noise ratio (CPSNR) of their outputs. Specifically, the CPSNR of a reconstructed full-color image I_r with respect to its original I_o is defined as

$$\mathrm{CPSNR} = 10\log_{10}\left(\frac{255^2}{\mathrm{CMSE}}\right) \tag{20}$$

where $\mathrm{CMSE} = \|I_o - I_r\|^2/3HW$ and $H \times W$ is the size of image I_o. The proposed algorithm provides the best performance.

Table II shows the performance of the evaluated methods in terms of CIELab color difference. The CIELab color difference is defined as the Euclidean distance between the original color of a pixel and its reproduction in CIELab color metric space [27]. Again, the proposed algorithm provides the best performance.

As mentioned earlier, using sophisticated edge-sensing interpolation algorithms instead of BI after demosaicking in the zooming-after-demosaicking approach may not provide a better zooming performance. Table III shows the zooming performance of some other combinations [28], [29] for comparison. It shows that BI is a good choice in view of both complexity and quality.

Objective measures may not be accurate and reliable enough to tell the quality difference among the processing results. Figs. 9 and 10 show, respectively, some zooming results of images 19 and 8 for visual comparison. They show that the proposed algorithm outstandingly preserves the image features with less color artifacts. For example, as shown in Fig. 9, the proposed algorithm can reproduce the fence texture with very little color artifacts but the others cannot. One can also see from Fig. 10 that, while most of the other methods totally destroy the letters on the wall in their outputs, the proposed algorithm can

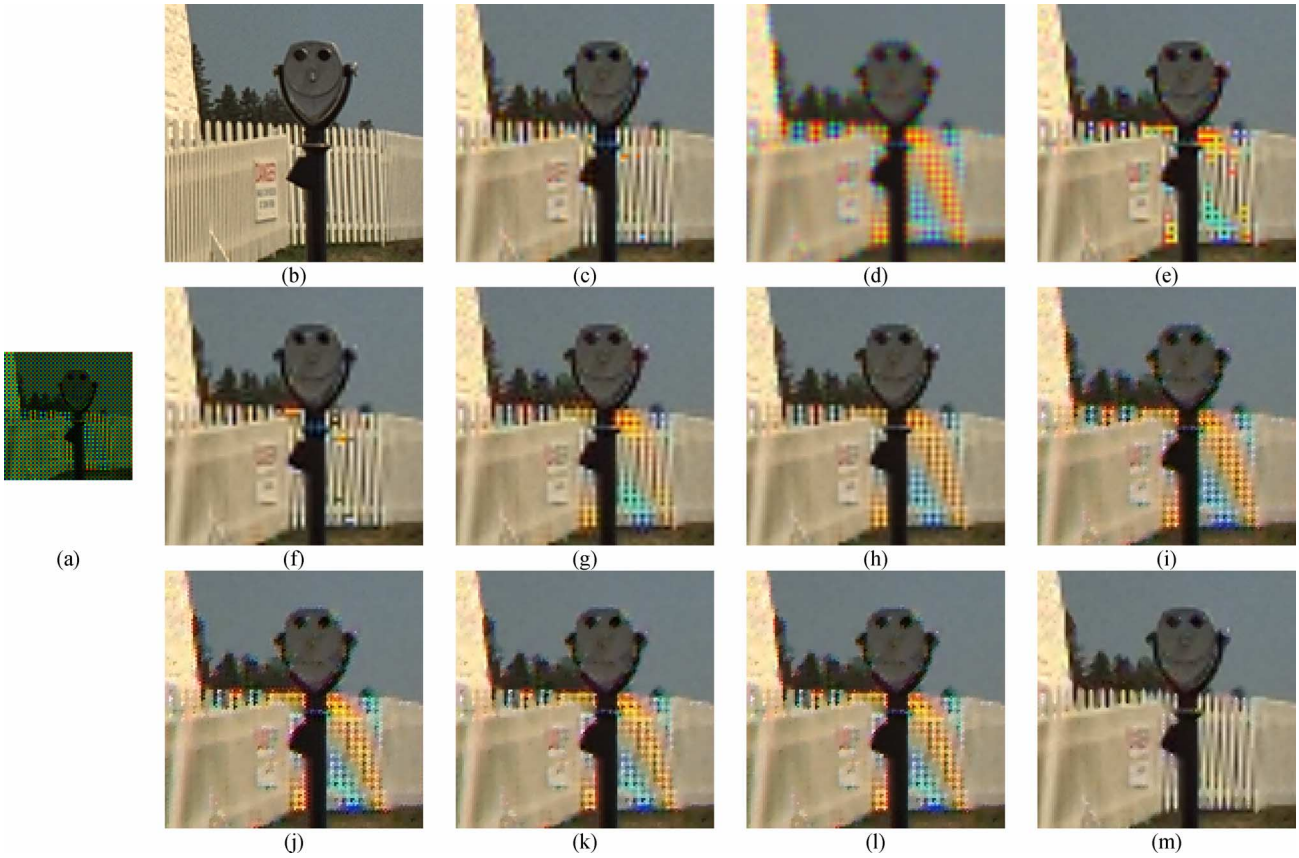


Fig. 9. Part of the processing results of image 19: (a) the input CFA image, (b) the full-color original, (c) PCSD+BI, (d) BICD+BI, (e) ACPI+BI, (f) AHDA+BI, (g) AP+BI, (h) NCED+BI, (i) DCZPS, (j) CIZBP+AP, (k) CIZBP+AHDA, (l) CIZBP+NCED, and (m) the proposed algorithm.

Fig. 10. Part of the processing results of image 8: (a) the input CFA image, (b) the full-color original, (c) PCSD+BI, (d) BICD+BI, (e) ACPI+BI, (f) AHDA+BI, (g) AP+BI, (h) NCED+BI, (i) DCZPS, (j) CIZBP+AP, (k) CIZBP+AHDA, (l) CIZBP+NCED, and (m) the proposed algorithm.

TABLE IV
COMPLEXITY REQUIRED FOR ESTIMATING A MISSING COMPONENT IN DIFFERENT STAGES OF THE PROPOSED JOINT DEMOSAICKING AND ZOOMING ALGORITHM

Action taken in a particular stage			Operation			
Action	**Stage**		**ADD**	**MUL**	**CMP**	**ABS**
Estimating a missing G component	(a) → (c) in Fig.7	In sharp edge region	33	3	2	8
		Not in sharp edge region	89	31	4	8
	(c) → (d) in Fig.7		30	5	5	6
	(d) → (e) in Fig.7		3	1	0	0
Estimating a missing R or B component	(e) → (f) in Fig.7		2	2	0	0
	(f) → (g) in Fig.7		2	2	0	0

preserve most of the details and introduces less color artifacts. These results, to a large extent, reflect the robustness of the proposed joint demosaicking and zooming algorithm in which the interpolation direction is determined directly from the raw sensor data and then used effectively and consistently in different stages to interpolate the green plane.

CIZBP [22] was a dedicated algorithm proposed for zooming CFA images. To make a direct comparison with CIZBP, zoomed CFA images produced by the proposed algorithm were extracted from the intermediate results having the available color components shown in Fig. 7(f) for comparison. Corresponding reference CFA images were generated by subsampling I_o according to the Bayer pattern. The difference between the zoomed CFA image output of a particular algorithm and its corresponding reference was then measured in terms of $\mathrm{PSNR} = 10\log_{10}((255)^2/\mathrm{MSE})$, where MSE is the mean square error of all available color components in the involved CFA image. On average, the PSNRs achieved by CIZBP and the proposed algorithm are 25.51 and 28.17 dB, respectively.

V. COMPUTATIONAL COMPLEXITY

In this section, a computational complexity analysis of the proposed algorithm is given, in which the complexity is measured in terms of number of operations including addition (ADD), multiplication (MUL), bit-shift (SHT), comparison (CMP), and taking absolute value (ABS).

Table IV summarizes the complexity required by the proposed algorithm to estimate a missing component in different stages. The stages into which the algorithm is decomposed in our complexity analysis matches those presented in Fig. 7 for easy reference. Note that some intermediate computation results can be reused in later stages and this was taken into account when the complexity of the proposed algorithm was estimated.

In practice, the real number of operations required to produce a zoomed full-color image with the proposed algorithm is image-dependent and it relies on how many missing green components are in sharp edge regions. Table V lists the average number of operations per pixel required by different methods in processing 24 testing images in our simulations. The complexity of BICD+BI and ACPI+BI is lower than that of the proposed algorithm but their output quality is poor as shown in Figs. 9 and 10. As for PCSD+BI, its complexity is similar to ours but its output quality is a bit lower especially in terms of the CIELab color difference criterion. The complexity of the proposed algorithm is around 25% of that of DCZPS [23]. When zooming a CFA image, the complexity of the proposed algorithm is around 67% of that of CIZBP [22]. In fact, the computation effort for the proposed algorithm to produce a zoomed full-color image is less than that required for CIZBP [22] to produce a zoomed CFA image.

TABLE V
AVERAGED NUMBER OF OPERATIONS PER PIXEL REQUIRED BY VARIOUS ALGORITHMS

Methods	**ADD**	**MUL**	**CMP**	**SHT**	**ABS**	**Total**
Producing a zoomed full-color image						
BICD+BI	5.25	0.00	0.00	4.25	0.00	**9.50**
ACPI+BI	5.87	0.00	0.38	3.49	0.99	**10.73**
PCSD+BI	25.00	1.50	0.75	3.50	3.00	**33.75**
AHDA+BI	16.25	2.00	9.75	8.25	1.00	**37.25**
AP+BI	64.77	7.88	0.13	32.25	0.50	**105.53**
NCED+BI	71.25	34.50	0.00	10.25	2.00	**118.00**
DCZPS	67.50	43.44	0.00	0.00	16.88	**127.81**
Ours	21.53	4.19	2.75	2.91	2.43	**33.82**
Producing a zoomed CFA image						
CIZBP	25.88	14.25	0.00	0.00	4.50	**44.63**
Ours	18.78	4.19	2.75	1.54	2.43	**29.69**

In our simulations, the average execution time for the proposed algorithm to produce a zoomed full-color image from a 256×384 CFA image with a zooming factor of 2 on a 2.8-GHz Pentium 4 PC with 512-MB RAM is 0.0774 s.

VI. CONCLUSION

In this paper, a low complexity joint demosaicking and zooming algorithm is proposed. With the use of the local color-difference variances, the raw sensor data is considered directly to determine the interpolation direction for estimating the missing green components in the zoomed image. With this arrangement, the green plane can be efficiently interpolated with preserved image details. With reference to the interpolated green plane, the red and the blue planes are then interpolated at low complexity. Simulation results show that the proposed algorithm produces images providing the most details and the least color artifacts at the lowest complexity as compared with conventional approaches which generally perform demosaicking after zooming or zooming after demosaicking.

ACKNOWLEDGMENT

The authors would like to thank Dr. B. K. Gunturk and Dr. R. Lukac for providing programs of their algorithms [5], [8], [22], [23].

REFERENCES

[1] R. Lukac and K. N. Plataniotis, "Color filter array: Design and performance analysis," *IEEE Trans. Consum. Electron.*, vol. 51, no. 4, pp. 1260–1267, Nov. 2005.

[2] B. E. Bayer, "Color imaging array," U.S. Patent 3 971 065, Jul. 1976.

[3] W. T. Freeman, "Median filter for reconstructing missing color samples," U.S. Patent 4 724 395, 1988.

[4] J. F. Hamilton and J. E. Adams, "Adaptive color plane interpolation in single sensor color electronic camera," U.S. Patent 5 629 734, 1997.

[5] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," *IEEE Trans. Image Process.*, vol. 11, no. 9, pp. 997–1013, Sep. 2002.

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

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

[8] R. Lukac, K. N. Plataniotis, D. Hatzinakos, and M. Aleksic, "A novel cost effective demosaicking approach," *IEEE Trans. Consum. Electron.*, vol. 50, no. 1, pp. 256–261, Feb. 2004.

[9] T. Sakamoto, C. Nakanishi, and T. Hase, "Software pixel interpolation for digital still camera suitable for a 32-bit MCU," *IEEE Trans. Consum. Electron.*, vol. 44, no. 4, pp. 1342–1352, Nov. 1998.

[10] S. C. Pei and I. K. Tam, "Effective color interpolation in CCD color filter arrays using signal correlation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 6, pp. 503–513, Jun. 2003.

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

[12] S. Battiato, G. Gallo, and F. Stanco, "A locally adaptive zooming algorithm for digital images," *Image Vis. Comput.*, vol. 20, no. 11, pp. 805–812, Sep. 2002.

[13] P. Thevenaz, T. Blu, and M. Unser, "Interpolation revisited," *IEEE Trans. Med. Imag.*, vol. 19, no. 7, pp. 739–758, Jul. 2000.

[14] J. W. Hwang and H. S. Lee, "Adaptive image interpolation based on local gradient fearutres," *IEEE Signal Process. Lett.*, vol. 11, no. 3, pp. 356–362, Mar. 2004.

[15] L. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2226–2238, Aug. 2006.

[16] N. Herodotou and A. N. Venetsanopoulos, "Colour image interpolation for high resolution acquisition and display devices," *IEEE Trans. Consum. Electron.*, vol. 41, no. 4, pp. 1118–1126, Nov. 1995.

[17] R. Lukac, B. Smolka, K. Martin, and K. N. Plataniotis, "Vector filtering for color imaging," *IEEE Signal Process. Mag.*, vol. 22, no. 1, pp. 74–86, Jan. 2005.

[18] R. Lukac, K. N. Plataniotis, B. Smolka, and A. N. Venetsanopoulos, "Vector operators for color image zooming," in *Proc. IEEE Int. Symp. Industrial Electronics ISIE*, Dubrovnik, Croatia, Jun. 2005, vol. 3, pp. 1273–1277.

[19] Y. Cha and S. Kim, "Edge-forming methods for color image zooming," *Proc. IEEE*, vol. 15, no. 8, pp. 2315–2323, Aug. 2006.

[20] P. Longere, X. Zhang, P. B. Delahunt, and D. H. Brainard, "Perceptual assessment of demosaicking algorithm performance," *Proc. IEEE*, vol. 90, no. 1, pp. 123–132, Jan. 2002.

[21] R. Lukac and K. N. Plataniotis, "Digital zooming for color filter array based image sensors," *Real-Time Imag.*, vol. 11, no. 2, pp. 129–138, Apr. 2005.

[22] R. Lukac, K. N. Plataniotis, and D. Hatzinakos, "Color image zooming on the Bayer pattern," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 11, pp. 1475–1492, Nov. 2005.

[23] R. Lukac, K. Martin, and K. N. Plataniotis, "Digital camera zooming based on unified CFA image processing steps," *IEEE Trans. Consum. Electron.*, vol. 50, no. 1, pp. 15–24, Feb. 2004.

[24] D. Taubman, "Generalized wiener reconstruction of images from colour sensor data using a scale invariant prior," in *Proc. Int. Conf. Image Processing*, Vancouver, BC, Canada, Sep. 10–13, 2000, pp. 801–804.

[25] J. Adams, "Design of practical color filter array interpolation algorithms for digital cameras," *Proc. SPIE*, vol. 3028, pp. 117–125, Feb. 1997.

[26] R. H. Hibbard, "Apparatus and method for adaptively interpolation a full-color image utilizing luminance gradients," U.S. Patent 5 382 976, 1995.

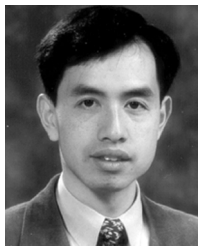
[27] K. McLaren, "The development of the CIE 1976 (L*a*b*) uniform color-space and colour-difference formula," *J. Soc. Dyers Color.*, vol. 92, pp. 338–341, 1976.

[28] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, no. 12, pp. 1153–1160, Dec. 1981.

[29] S. Battiato, G. Gallo, and F. Stanco, "A new edge-adaptive zooming algorithm for digital camera," in *Proc. Signal Processing Communications*, Marbella, Spain, 2000, pp. 144–149.

King-Hong Chung received the B.Eng. (Hons.) degree in electronic and information engineering from The Hong Kong Polytechnic University in 2001, where he is currently pursing the Ph.D. degree.

His current research interests include digital image halftoning, image compression, and digital camera image processing.



Yuk-Hee Chan (M'92) received the B.Sc. degree (with honors) in electronics from Chinese University of Hong Kong in 1987 and the Ph.D. degree in signal processing from The Hong Kong Polytechnic University in 1992.

Between 1987 and 1989, he was an R&D Engineer at Elec & Eltek Group, Hong Kong. He joined The Hong Kong Polytechnic University in 1992 and is now an Associate Professor in the Department of Electronic and Information Engineering. He has published over 110 research papers in various international journals and conferences. His research interests include image and video compression, image restoration, halftoning, demosaicking, and fast computational algorithms in digital signal processing.

Dr. Chan is a member of the IEE. He was the Chairman of the IEEE Hong Kong Joint Chapter of CAS and COM in 2003 and 2004.