Region-adaptive Demosaicking with Weighted Values of Multidirectional Information

Jia Shi, Chengyou Wang, and Shouyi Zhang
School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai 264209, China
Email: {shijiasdu, zhangshouyisdu}@gmail.com; wangchengyou@sdu.edu.cn

Abstract —In this paper a region-adaptive demosaicking algorithm with low computational complexity for single-sensor digital cameras is proposed. The proposed algorithm firstly divides the input image into two kinds of regions and then adopts different interpolation methods for each type. The proposed interpolation method makes full use of bilinear's fast execution speeds in the smooth region. And it directly extracts and recovers edge information with weighted values of multidirectional components in edge regions. Experimental results show that the proposed method has an outstanding performance not only in subjective visual quality but also in terms of composite peak signal to noise ratio (CPSNR).

Index Terms—adaptive demosaicking, Bayer pattern, color filter array (CFA), smooth region, weighted average

I. Introduction

To simplify the process and consider the cost savings, digital cameras and video cameras usually use a single image sensor (e.g., CCD or CMOS). Their surface is covered by a layer of color filter array (CFA), which could only receive one kind of base shade at each pixel. For getting a full-color image, adopting an appropriate interpolation method, called demosaicking algorithm, at each point to recover the other two color components is necessary. The most widely used model is Bayer CFA sample array, shown in Fig. 1 [1]. Since human visual system is more sensitive to the green (G), Bayer sets that the number of green pixels is twice as the red (R)'s or the blue (B)'s.

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

Fig. 1. Color filter array (Bayer pattern).

The original Bayer CFA demosaicking algorithms are: nearest neighbor interpolation and bilinear interpolation [2], [3]. Since the computational complexity of these algorithms is at a low level, they have faster execution speeds. Computational complexity here measured by the times of addition and multiplication in an independent algorithm. However, the phenomenon of distortion at the edges of the image is distinct. To mitigate this problem around line edges, several demosaicking methods [4], [5] have been proposed, which first accurately identifies line edges with edge indicators and then estimates missing pixels with an edge-adaptive methods. Adams and Hamilton comprehensively considered chrominance (R or B) and luminance (G) information within 5×5 neighborhoods when calculating the horizontal and vertical gradient operators [6]. Missing color values in Lee's approach are estimated by using the additional information in 45° and 135° directions [7]. Wang and Lin improved edge detection in [8] by using surrounding pixels' values as well as employing information to get final edge direction of current pixel. Their work separated edge regions and other regions, which inspired our work so much.

Moriaan color model indicates that the ratio of each color component in a full-color image is almost constant [9]. Based on this model, several algorithms were proposed [10]-[12]. An adaptive filtering for color filter array demosaicking is proposed [10]. In order to reduce the mutual interference between the chrominance, an adaptive least squares inverse filtering method is proposed in [11], but the influence of different gradients of image restoration is ignored. Chung proposed a low-complexity joint color demosaicking and zooming algorithm in [12]. In this method, the interpolation of all missing red and blue components can be done in parallel, so the processing time can be saved. More recently, Mairal [13] and Yu [14] proposed demosaicking methods based on sparse representation of images. These algorithms assumed that patches in natural images admit a sparse representation over a dictionary.

We summarize the recent methods by Getreuer [15], [16] and Kiku [17], [18] being able to give state-of-the-art results in both databases. Getreuer's demosaicking algorithm is based on total variation along curves, and first estimates the image contour orientations directly from the mosaicked data using contours stencils. The demosaicking is performed as an energy minimization, using a graph regularization adapted according to the orientation estimates. The objective energy functional consists in two terms. The first one regularizes the luminance to suppress zipper artifacts while the second

Manuscript received August 24, 2014; revised November 25, 2014.
This work was supported by the National Natural Science Foundation of China (Grant No. 61201371) and the promotive research fund for excellent young and middle-aged scientists of Shandong Province, China (Grant No. BS2013DX022).
Corresponding author email: wangchengyou@sdu.edu.cn.
doi:10.12720/jcm.9.12.930-936

term regularizes the chrominance to suppress color artifacts. Kiku proposed a strategy in [17] that consists in the interpolation of the residual differences, which means the differences between observed and tentatively estimated pixel values. Fan [19] proposed a constant-hue-based color filter array demosaicking sensor for digital still camera implementation. Chen [20] proposed an efficient post-processing method to reduce interpolation artifacts based on the color difference planes.

This paper proposes a region-adaptive method with weighted values of multidirectional information. Different from conventional interpolation methods based on two directions or four directions, the proposed method exploits greater degree correlations among neighboring pixels along eight directions to improve the interpolation performance. We identify region types (smooth region or edge region) by gradient values, then choose different treatment for different areas: In the smooth region, use bilinear interpolation which has obvious advantages in computational complexity aspect; in the edge region, take multidirectional pixel information into consideration by employing weighted gradient values. This algorithm's region-adaptive idea and time-saving superiority are inspired by [8] and [12], and it circumvents the bad effect during image restoration caused by different gradients which has appeared in [11]. By comparing with methods in related literatures, the algorithm has better performance in the recovery of the green component and reconstruction of the overall image.

The remainder of this paper is organized as the following. Section II is devoted to the adaptive demosaicking algorithm. Section III introduces the proposed interpolation method. Experimental results are presented with other existing methods in Section IV, and conclusions are provided in Section V.

II. Adaptive Demosaicking

Biologically speaking, the human visual system is sensitive to sudden changes of the color and edge information. So efficient interpolation algorithms are almost combined with edge and texture information. Since the number of green component occupies half of whole pixels in Bayer array, interpolation algorithm generally gives priority to restore the green component.

To deal with the difference between the edge and texture, adaptive demosaicking method has been proposed. When recovering the green component, firstly calculate the gradient operators in different directions, and then select the appropriate interpolation direction.

As Fig. 2 shows, G_i represents the green component, while A_i stands for red or blue component. All A_i pixels will be the same color for the entire neighborhood. For simplification, we will use the term chrominance to represent either red or blue. We define operators in horizontal direction and vertical direction as D_{H} and D_{V} respectively.

		A_1		
		G_2		
A_3	G_4	A_5	G_6	A_7
		G_8		
		A_9		

Fig. 2. Bayer Pattern neighborhood.

Hibbard detects edges by calculating first-order differential [4]:

$$D_{\mathrm{H}} = |G_4 - G_6| \tag{1}$$

$$D_{\mathrm{V}} = |G_2 - G_8| \tag{2}$$

Laroche proposes second-order terms [5]:

$$D_{\mathrm{H}} = |-A_3 + 2A_5 - A_7| \tag{3}$$

$$D_{\mathrm{V}} = |-A_1 + 2A_5 - A_9| \tag{4}$$

Adams and Hamilton improve operators on the basis of the above [6]:

$$D_{\mathrm{H}} = |-A_3 + 2A_5 - A_7| + |G_4 - G_6| \tag{5}$$

$$D_{\mathrm{V}} = |-A_1 + 2A_5 - A_9| + |G_2 - G_8| \tag{6}$$

The minimum one is chosen as the preferred orientation for the interpolation. The details are as follows [6].

$$G_5 = \begin{cases} \frac{1}{2}(G_2 + G_8), \ D_{\mathrm{V}} < D_{\mathrm{H}} \\ \frac{1}{4}(G_2 + G_4 + G_6 + G_8), \ D_{\mathrm{V}} = D_{\mathrm{H}} \\ \frac{1}{2}(G_4 + G_6), \ D_{\mathrm{V}} > D_{\mathrm{H}} \end{cases} \tag{7}$$

The second pass of the interpolation fully populates the red and blue color planes. Consider the following neighborhood in Fig. 3.

A_1	G_2	A_3
G_4	C_5	G_6
A_7	G_8	A_9

Fig. 3. 3×3 chrominance neighborhood.

G_i is a green pixel, A_i is either a red or blue pixel and C_5 is the opposite color pixel to A_i (i.e., if A_i is red then C_5 is blue and vice versa). Here we assume that all G_i has been known.

There are three cases [6]. Case 1 is when the nearest neighbors to A_i are in the same column. (A_4 is used as an example)

$$A_4 = (A_1 + A_7)/2 + (-G_1 + 2G_4 - G_3)/2 \tag{8}$$

Case 2 is when the nearest neighbors to A_i are in the same row. (A_2 is taken as an example)

$$A_2 = (A_1 + A_3)/2 + (-G_1 + 2G_2 - G_3)/2 \tag{9}$$

Case 3 is when the nearest neighbors to A_i are at the four corners (A_5 is taken as an example).

$$A_5 = (A_1 + A_3 + A_7 + A_9)/4 + (-G_1 - G_3 + 4G_5 - G_7 - G_9)/4 \quad (10)$$

Beside this way to recover R or B components, there is another way to treat chrominance plane interpolation. The color difference model used is employed in [12] when the missing red and blue components are constituted. Its green-to-red (green-to-blue) color difference value is bilinearly interpolated from the neighboring pixels in which red (blue) CFA components are already known, and its intensity value can then be determined.

For example, when the red components of pixels (i, j), $(i, j+4)$, $(i+4, j)$ and $(i+4, j+4)$ are known and the value of $(i+m, j+n)$ is waiting to be estimated, where $0 \le m,\ n \le 4$. The green-to-red color difference value of pixel $(i+m, j+n)$ is first interpolated as the following:

$$\begin{aligned} d_{(g-r)(i+m,j+n)} = {} & \frac{4-n}{4}\left[\frac{m}{4}d_{(g-r)(i+4,j)} + \frac{4-m}{4}d_{(g-r)(i,j)}\right] \\ & + \frac{n}{4}\left[\frac{m}{4}d_{(g-r)(i+4,j+4)} + \frac{4-m}{4}d_{(g-r)(i,j+4)}\right],\ 0 \le i, j \le 4. \end{aligned} \quad (11)$$

The missing red color component is then estimated by

$$R_{(i+m,j+n)} = G_{(i+m,j+n)} - d_{(g-r)(i+m,i+n)}. \quad (12)$$

The missing blue color component is treated in the same way. In practice, the interpolation of red and blue components can be done in parallel so as to reduce the processing time.

The former algorithm only uses one of the horizontal or vertical directions of the gradient component, which completely ignores the constitution to recovery of the information from other directions. The latter method presents the integrated use of the information on the four corners within a 5×5 neighborhood.

III. Proposed Algorithm

To reach a better recovery performance with low computational complexity, the proposed scheme improved the original adaptive interpolation method introduced in Section II. In the field of data structure, computational complexity, this is also called algorithmic complexity, measured by the addition times as well as multiplication times. A relatively small computational complexity method would be favored since it means higher efficiency.

When using different demosaicking methods to reconstruct a same original image, the number of loops in their corresponding programs is completely equal. The reason is that whatever a method is, its goal is to estimate the two losing components for every pixel. Thereby the times that we use each method is 2 times of an image size. For example, in our work, the size of test images are 512×512 pixels. Fig. 4 shows six original 24-bit (8-bit for each color component) full-color images used in the simulation.

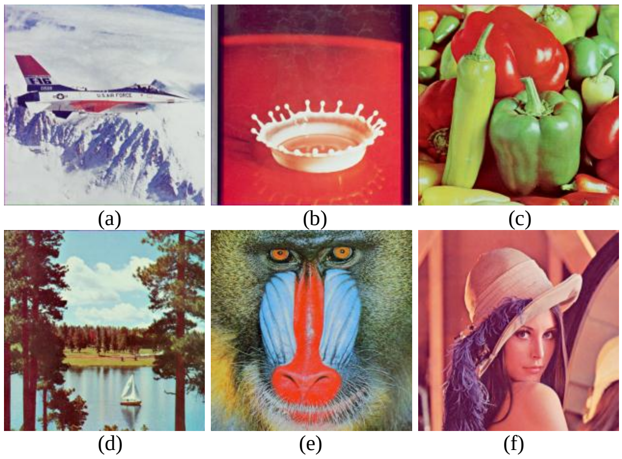


Fig. 4. Original full-color images: (a) Airplane, (b) Milkdrop, (c) Peppers, (d) Boat, (e) Mandrill, and (f) Lena.

The core part of each algorithms, the loop body, runs $2\times512\times512$ times in a real program. So, when we compared computational complexity of different methods, the comparison of their core part is enough. Table I shows comparison of different methods' core part's computational complexity.

TABLE I: COMPUTATIONAL COMPLEXITY OF DIFFERENT METHODS (CORE PART).

Method	Bilinear	ACP[6] +ADW[8]	LCGC [12]	CHB [19]	MDW [20]
Addition(times)	3	14	24	20	17
Multiplication (times)	1	4	4	4	5

Bilinear method's extraordinary low computational algorithm is ascribed to the following two reasons: First, it doesn't contain the process of justifying interpolation direction, and the second aspect is that it simply used the average value of pixels in four orientations (up, down, left, right).

In the green components recovery pass, the scheme scans the CFA image and detects if a particular pixel is in a smooth region. If it is, bilinear interpolation method will be adopted. Otherwise, the pixel will use weighted values of multidirectional information within its 5×5 neighborhood as the missing green component value. The same process is applied to the recovery of R/B components pass.

Fig. 5 outlines how to select an appropriate method for a particular pixel of the proposed scheme. The whole algorithm can be divided into two blocks: the first is the interpolation of green component, and the second part is towards red and blue components. The details are as the following.

A. *Region-adaptive Demosaicking*

Bilinear interpolation, one of the classic demosaicking methods, can assure a high quality of recovery in smooth region with the absolute advantage in speed. Inspired by Wang and Lin [8], we take different interpolation methods in smooth region and edge region, that is to say, once a pixel is justified in a smooth region, we use bilinear method, and when a pixel is in an edge region, we interpolate the missing colors with weighted values of multidirectional information.

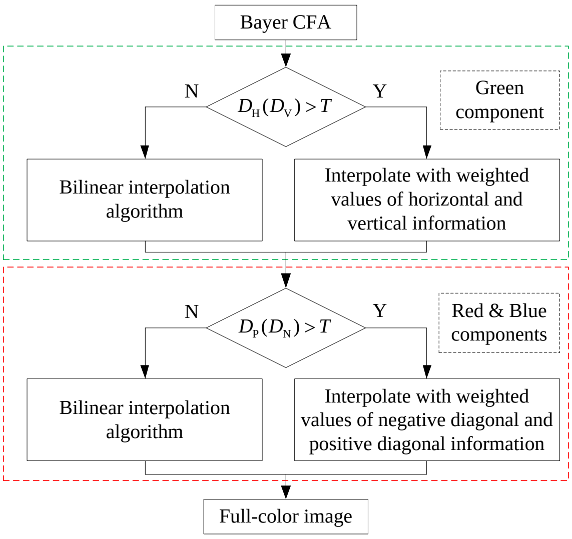


Fig. 5. Overview of the proposed demosaicking method.

A region's type (smooth region or edge region) is determined by its gradient operators, Eqs. (5) and (6) are applied for D_H, D_V in our proposed method:

$$|D_H - D_V| \le T \tag{13}$$

$$|D_H - D_V| > T \tag{14}$$

where T stands for the threshold to identify different region types. If gradient operators agree with (13), we consider it is in a smooth region. And (14) is the requirement for edge regions. Table II shows that the performance of composite peak signal to noise ratio (CPSNR) [21] and speed with respectively T value, and we conducted simulations using MATLAB with a processor of Intel(R) Core(TM) i5-2430 CPU, M380 @ 2.40GHz, RAM 8.00GB.

TABLE II: CPSNR AND SPEED PERFORMANCE WITH DIFFERENT T APPLIED TO IMAGE LENA.

T	1	2	3	4	5
CPSNR(dB)	35.5421	35.5034	35.4652	35.4332	**35.4035**
Time(s)	2.3211	2.2321	2.2208	2.1753	**2.0812**
T	6	7	8	9	10
CPSNR(dB)	35.3223	35.2475	35.1735	35.1064	35.0222
Time(s)	2.0820	2.0756	2.0715	2.0633	2.0548

Fig. 6 represents the trends of CPSNR and speed with different T. When T increases from 1 to 5, time declines obviously and CPSNR changes relatively flat. When T increases from 6 to 10, time is at a smooth state, and CPSNR, which reflects the reconstruct quality, declines rapidly. Since our aim is to find a T value which corresponds less time and at the same time maintains a high CPSNR, then 5 is a proper and ideal value.

Fig. 7 shows visual comparison of reconstructed Fig. 4(b) produced by demosaicking methods. The number of smooth region pixels of Fig. 4(b) is comparatively at a high level. We can see that the bilinear algorithm, compared with other interpolation methods, does a considerably good recovery in smooth regions. In this paper, considering the computational complexity and CPSNR two factors, we suppose T is 5.

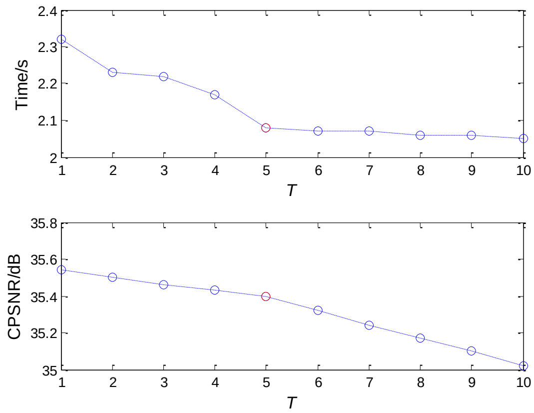


Fig. 6. The time and CPSNR with different T applied to image Lena.

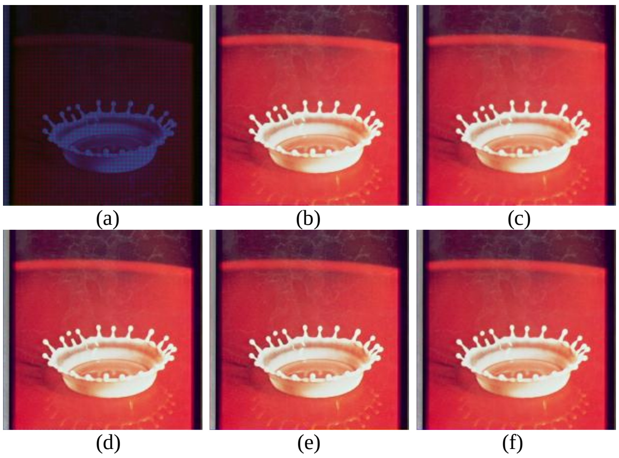


Fig. 7. The processing results of image Milkdrop: (a) the input CFA image, (b) the full-color original, (c) bilinear, (d) ACP+ADW, (e) LCGC, and (f) the proposed algorithm.

B. Weighted Values of Multidirectional Information

In the first step, when luminance information is restored, the weighting factor of how different directions operators effect on interpolation can be calculated as long as the horizontal and vertical gradient operators is calculated. Unlike the original algorithms to select a best interpolation direction, a weighted value of multidirectional information use more original green components when restore missing green components. The weighted values of horizontal direction W_H and vertical direction W_V can be calculated:

$$W_H = D_V / (D_H + D_V) \tag{15}$$

$$W_V = D_H / (D_H + D_V) \tag{16}$$

The complete green interpolation process now is expressed as below, considering the 3×3 neighborhood as shown in Fig. 3.

if $|D_{\mathrm{H}} - D_{\mathrm{V}}| > T$,

$$G_5 = [(G_2 + G_8)/2 + (2\times A_5 - A_1 - A_9)/4]\times W_{\mathrm{V}} + [(G_4 + G_6)/2 + (2\times A_5 - A_3 - A_7)/4]\times W_{\mathrm{H}}, \tag{17}$$

else

$$G_5 = (G_2 + G_4 + G_6 + G_8)/4 \tag{18}$$

The second step of the interpolation fully populates the red and blue color planes. Considering the following neighborhood in Fig. 8, operators in positive direction D_{P} and negative direction D_{N} are defined as the following:

$$D_{\mathrm{P}} = |-G_3 + 2G_5 - G_7| + |A_3 - A_7| \tag{19}$$

$$D_{\mathrm{N}} = |-G_1 + 2G_5 - G_9| + |A_1 - A_9| \tag{20}$$

A_1	G_2	A_3	G_4	A_5
G_6	C_7	G_8	C_9	G_{10}
A_{11}	G_{12}	A_{13}	G_{14}	A_{15}
G_{16}	C_{17}	G_{18}	C_{19}	G_{20}
A_{21}	G_{22}	A_{23}	G_{24}	A_{25}

Fig. 8. 5×5 Chrominance neighborhood.

This step is similar to the first step, when recover the chrominance information (R/B), calculate the weighted values of positive and negative directions (W_{P}, W_{N}):

$$W_{\mathrm{P}} = D_{\mathrm{N}} / (D_{\mathrm{N}} + D_{\mathrm{P}}) \tag{21}$$

$$W_{\mathrm{N}} = D_{\mathrm{P}} / (D_{\mathrm{N}} + D_{\mathrm{P}}) \tag{22}$$

The complete chrominance components interpolation process now is expressed as below:

if $|D_{\mathrm{N}} - D_{\mathrm{P}}| > T$,

$$A_{13} = [(A_1 + A_{25})/2 + (2\times G_{13} - G_7 - G_9)/4]\times W_{\mathrm{N}} + [(A_5 + A_{21})/2 + (2\times G_{13} - G_9 - G_{17})/4]\times W_{\mathrm{P}} \tag{23}$$

else

$$G_5 = (G_2 + G_4 + G_6 + G_8)/4 \tag{24}$$

IV. Experimental Results

Experiments were conducted in order to evaluate the performance of the proposed demosaicking algorithm. In this paper all simulation results are obtained with MATLAB 7.14. The original full-color images in Fig. 4 were subsampled according to the Bayer CFA pattern, with starting sampling sequence of "RGRG…" in the first row, to form a set of CFA testing images. The CFA testing images were then processed with bilinear, ACP [6]+ADW [8], LCGC [12], MDW [20], and proposed algorithm to produce full-color images for comparison. In all simulations, we adopted the point-symmetric boundary extension [22] to realize the prefect reconstruction in Bayer pattern.

Table III tabulates the performance of various methods in terms of the peak signal to noise ratio (PSNR) of green components between the input image and the reconstructed image. And Table IV tabulates the performance of different methods in terms of the CPSNR [21]. Specifically, the PSNR and CPSNR of a reconstructed full-color image are defined as

$$\mathrm{PSNR} = 10\log_{10}\left[\frac{255^2}{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[I_{\mathrm{in}}(i,j,2) - I_{\mathrm{out}}(i,j,2)\right]^2}\right] \tag{25}$$

and

$$\mathrm{CPSNR} = 10\log_{10}\left[\frac{255^2}{\frac{1}{3MN}\sum_{k=1}^{3}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[I_{\mathrm{in}}(i,j,k) - I_{\mathrm{out}}(i,j,k)\right]^2}\right] \tag{26}$$

where I_{in} is the input image, I_{out} is the output image, and $M\times N$ is the size of image. In Table III, We focus on the recovery of green components, since they play a fundamental role in the whole interpolation, in other words, the reconstruction of red and blue components are based on the green components interpolation. Thus a high PSNR on green component is a necessary precondition of the ideal whole demosaicking result. Both in PSNR of green component and CPSNR, the proposed algorithm provides the best performance (except PSNR of MDW on green components of Peppers).

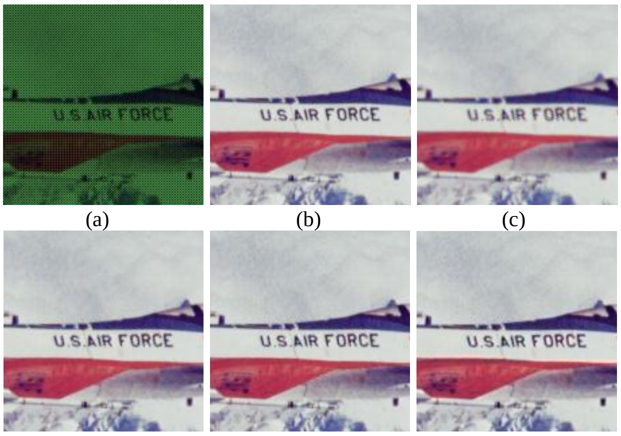
TABLE III: PSNR OF BILINEAR, ACP+ADW, LCGC, MDW, AND PROPOSED METHOD ON G COMPONENTS.

Image	Bilinear	ACP+ADW	LCGC	MDW	Proposed
Airplane	33.5748	36.2210	38.3583	38.6234	**38.7549**
Milkdrop	35.1312	37.5822	39.8912	40.1132	**40.3982**
Peppers	32.0472	33.6660	34.2934	34.4235	**34.3498**
Boat	29.7299	30.6362	30.9423	31.0423	**31.0462**
Mandrill	25.5471	26.8255	27.7663	27.8564	**28.0437**
Lena	35.4812	35.9496	37.4362	37.7337	**37.8233**
Average	31.9186	33.4801	34.7813	34.9654	**35.0694**

TABLE IV: CPSNR OF BILINEAR, ACP+ADW, LCGC, MDW, AND PROPOSED METHOD.

Image	Bilinear	ACP+ADW	LCGC	MDW	Proposed
Airplane	32.4525	35.4533	37.8767	37.6334	**37.8867**
Milkdrop	34.3245	36.7645	38.6466	39.0189	**39.2111**
Peppers	31.3452	32.6527	33.6842	33.7235	**34.0492**
Boat	29.2347	30.3454	29.8876	30.0483	**30.0662**
Mandrill	24.7345	26.8742	26.3432	27.1564	**27.3433**
Lena	34.3457	35.0446	35.1218	36.0337	**36.4235**
Average	31.0729	32.8558	33.5934	33.9357	**34.1633**

Objective measures may not be accurate and reliable enough to illustrate the quality difference among the processing results. Fig. 9 shows visual comparison of reconstructed images produced by demosaicking methods. In Fig. 9, the proposed algorithm outstandingly preserves the letters on the airplane with less color artifacts in image Airplane.



(a) (b) (c)

(d) (e) (f)

Fig. 9. Part of the processing results of image Airplane: (a) the input CFA image, (b) the full-color original, (c) bilinear, (d) ACP+ADW, (e) LCGC, and (f) the proposed algorithm.

V. Conclusion

In this paper, a region-adaptive demosaicking with weighted values of multidirectional information is presented. With the use of weighted values, more components from original image are considered. Since bilinear interpolation can assure high quality of recovery in smooth region with the absolute advantage in speed, if we justify the region belongs to a smooth type, bilinear interpolation method is adopted. While an edge region will use the weighted value mentioned above. Simulation results show that the proposed algorithm produces images providing the most details and the least color artifacts with low level computational complexity.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 61201371) and the promotive research fund for excellent young and middle-aged scientists of Shandong Province, China (Grant No. BS2013DX022). The authors would like to thank the anonymous reviewers and the editor for their valuable comments to improve the presentation of the paper.

References

[1] B. E. Bayer, “Color imaging array,” U.S. Patent 3971065, Jul. 20, 1976.

[2] J. E. Adams, “Intersections between color plane interpolation and other image processing functions in electronic photography,” in *Proc. SPIE, Cameras and Systems for Electronic Photography and Scientific Imaging*, San Jose, CA, USA, Feb. 8-9, 1995, vol. 2416, pp. 144-151.

[3] H. S. Hou and H. C. Andrews, “Cubic spline for image interpolation and digital filtering,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 6, pp. 508-517, Dec. 1978.

[4] R. H. Hibbard, “Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients,” U.S. Patent 5382976, Jan. 17, 1995.

[5] C. A. Laroche and M. A. Prescott, “Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients,” U.S. Patent 5373322, Dec. 13, 1994.

[6] J. E. Adams and J. F. Hamilton, “Adaptive color plane interpolation in single sensor color electronic camera,” U.S. Patent 5629734, May 13, 1997.

[7] J. Lee, T. Jeong, and C. Lee, “Improved edge-adaptive demosaicking method for artifact suppression around line edges,” in *Proc. IEEE International Conference on Consumer Electronics*, Las Vegas, NV, USA, Jan. 10-14, 2007, pp. 1-2.

[8] S. Wang and F. Lin, “Adaptive demosaicking with improved edge detection,” in *Proc. IEEE International Workshop on Imaging Systems and Techniques*, Hong Kong, China, May 11-12, 2009, pp. 202-204.

[9] P. T. Eliason, L. A. Soderblom, and P. S. Chavez, “Extraction of topographic and spectral albedo information from multispectral images,” *Photogrammetric Engineering and Remote Sensing*, vol. 47, no. 11, pp. 1571-1579, Nov. 1981.

[10] N. X. Lian, L. L. Chang, Y. P. Tan, and V. Zagorodnov, “Adaptive filtering for color filter array demosaicking,” *IEEE Transactions on Image Processing*, vol. 16, no. 10, pp. 2515-2525, Oct. 2007.

[11] X. L. Wu and X. J. Zhang, “Joint color decrosstalk and demosaicking for CFA cameras,” *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3181-3189, Dec. 2010.

[12] K. H. Chung and Y. H. Chan, “A low-complexity joint color demosaicking and zooming algorithm for digital camera,” *IEEE Transactions on Image Processing*, vol. 16, no. 7, pp. 1705-1715, Jul. 2007.

[13] J. Mairal, M. Elad, and G. Sapiro, “Sparse representation for color image restoration,” *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 53-69, Jan. 2008.

[14] G. S. Yu, G. Sapiro, and S. Mallat, “Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity,” *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2481-2499, May 2012.

[15] P. Getreuer, “Color demosaicing with contour stencils,” in *Proc. 17th International Conference on Digital Signal Processing*, Corfu, Greece, Jul. 6-8, 2011, pp. 1-6.

[16] P. Getreuer, “Contour stencils: Total variation along curves for adaptive image interpolation,” *SIAM Journal on Imaging Sciences*, vol. 4, no. 3, pp. 954-979, Aug. 2011.

[17] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, “Residual interpolation for color image demosaicking,” in *Proc. IEEE International Conference on Image Processing*, Melbourne, VIC, Australia, Sep. 15-18, 2013, pp. 2304-2308.

[18] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, “Minimized-Laplacian residual interpolation for color image demosaicking,” in *Proc. SPIE, IS and T Electronic Imaging - Digital Photography X*, San Francisco, CA, USA, Feb. 3-5, 2014, vol. 9023, pp. 1-8.

[19] Y. C. Fan, Y. F. Chiang, and Y. T. Hsieh, “Constant-hue-based color filter array demosaicking sensor for digital still camera implementation,” *IEEE Sensors Journal*, vol. 13, no. 7, pp. 2586-2594, Jul. 2013.

[20] X. D. Chen, L. W. He, G. Jeon, and J. Jeong, “Multidirectional weighted interpolation and refinement method for Bayer pattern CFA demosaicking,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 8, pp. 1271-1282, Aug. 2015.

[21] J. Mukherjee, M. K. Lang, and S. K. Mitra, “Demosaicing of images obtained from single-chip imaging sensors in YUV color space,” *Pattern Recognition Letters*, vol. 26, no. 7, pp. 985-997, May 2005.

[22] Z. X. Hou, C. Y. Wang, and A. P. Yang, “Study on symmetric extension methods in Mallat algorithm of finite length signal,” in *Proc. 5th International Conference on Visual Information Engineering*, Xi'an, China, Jul. 29-Aug. 1, 2008, pp. 341-346.

Jia Shi was born in Shandong province, China in 1994. She was admitted into the School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai, China, in 2011. Now she is a fourth year student and her major course is software engineering. Her current research interest is image processing.

Chengyou Wang was born in Shandong province, China in 1979. He received his B.E. degree in electronic information science and technology from Yantai University, China, in 2004 and his M.E. and Ph.D. degree in signal and information processing from Tianjin University, China, in 2007 and 2010 respectively. Now he is an associate professor in the School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai, China. His current research interests include image processing and transmission technology, multidimensional signal and information processing, and smart grid technology.

Shouyi Zhang was born in Shandong province, China in 1993. He was admitted into the School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai, China, in 2011. Now he is a fourth year student and his major course is communication engineering. His current research interest is image processing and signal processing.