Iterative Enhanced Directional Based Demosaicking Algorithm

A.Jayachandran[#] , Dr.R.Dhanasekaran*, Prof P. Rajan[#]

[#] PSN College of Engineering and Technology ,Tirunenveli,India
[*] Principal , Syed Ammal Engineering college,Ramnad,India
ajaya1675@gmail , rdhanashekar@yahoo.com, prajan1968@gmail.com

ABSTRACT

Demosaicing is a process of obtaining a full color image by interpolating the missing colors of an image captured from a digital still and video cameras that use a single-sensor array. This paper provides an effective iterative demosaicing algorithm applying a weighted-edge interpolation to handle green pixels followed by a series of color difference interpolation to update red, blue, and green pixels. Then border of the image is enhanced using some adaptive techniques. Experimental results show that the proposed method performs much better than other latest demosaicing techniques in terms of image quality. In comparison to the other algorithms, this algorithm proposed here results in increased PSNR value.

Keywords- Bayer's sampling, Iterative demosaicing, color interpolation, color difference

I. INTRODUCTION

In order to reduce the hardware cost, many digital still cameras (DSC) use a single sensor equipped with a color filter array (CFA) to capture any of the three primary color components, *R* (Red), *G* (Green), or *B* (Blue), on each pixel location. Among the various suggested CFAs,the Bayer CFA pattern [3] shown in Fig.1 is the most prevalent one, where *G* pixels occupy half of all, and *R* and *B* pixels share the others. A representation of a full-color image needs all the information from the three colors on each pixel location. As a result, the missing two colors on each pixel location have to be interpolated back to get a full-color image. The process of interpolating the missing colors is called as demosaicing or color interpolation whose main objective aims to reconstruct the missing colors as closely to the original ones as possible while keeping the computational complexity as low as possible.

The design and performance characteristics of the CFA are essentially determined by the type of a color system and the arrangements of the color filters in the CFA The visual effect of an RGB color images is based on the weight given to the RGB components

A demosaicing algorithm can be either heuristic or nonheuristic. A heuristic approach does not try to solve a mathematically defined optimization problem while a non- heuristic approach does. Most existing demosaicing

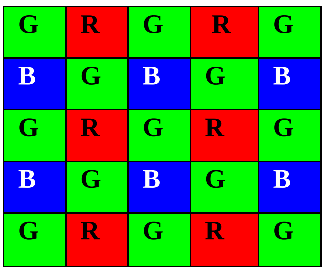


Fig-1, Bayer CFA pattern

algorithms are heuristic algorithms. It can be found that a number of heuristic algorithms were developed based on the framework of the adaptive color plane interpolation algorithm (ACPI) proposed in [2]. In this paper, based on the framework of ACPI, a new heuristic demosaicing algorithm is proposed. This algorithm uses the variance of pixel color differences to determine the interpolation direction for interpolating the missing green samples. Simulation results showed more noises in the border of the image ,so the border of the image is enhanced using some adaptive techniques. Now Simulation results show that the proposed algorithm is superior to the latest demosaicing algorithms in terms of increased PSNR value.

The paper is organized as follows. Section II presents the details of our demosaicing algorithm, and Section III presents some simulation results for comparison study. Finally, a conclusion is given in Section IV.

II **PROPOSED ALGORITHM**

Based on the observations in [6], the proposed algorithm put its focus on how to effectively determine the interpolation direction for estimating a missing green component in edge regions and texture regions. In particular, variance of color differences is used in the proposed algorithm as a criterion to determine the interpolation direction for the green components. This proposed algorithm is shown in fig.2

A. Interpolating Missing Green Components

Interpolating Missing Green Components

The color image has three color plane Red plane ,Green plane and Blue Plane The missing green plane is calculate first ,the remaining color plane are calculated Based on green plane ,ie the algorithm first calculate the two different planes $K_R \equiv G-R$ and $K_B \equiv G-B$ .In each missing green samples is first calculated the Weighted color difference value around the original pixel then to calculate the average of its neighboring color difference value .

Figure 2 The proposed CFA pattern having their centers at Red CFA samples. As shown in Figure 2, to obtain the $G_{i,j}$ value at the $R_{i,j}$ pixel ,first to calculate the weights along the four adjacent direction as follows

$$\begin{aligned}
\alpha_{i-1,j} &= |R_{i-2,j} - R_{i,j}| + |G_{i-1,j} - G_{i+1,j}| \\
\alpha_{i+1,j} &= |R_{i+2,j} - R_{i,j}| + |G_{i-1,j} - G_{i+1,j}| \\
\alpha_{i,j-1} &= |R_{i,j-2} - R_{i,j}| + |G_{i,j-1} - G_{i,j+1}| \qquad (1) \\
\alpha_{i,j+1} &= |R_{i,j+2} - R_{i,j}| + |G_{i,j-1} - G_{i,j+1}|
\end{aligned}$$

Calculate The Four adjacent color difference is calculated as follows

$$\begin{aligned}
Kr_{i-1,j} &= G_{i-1,j} - 1/2\,(R_{i-2,j} + R_{i,j}) \\
Kr_{i+1,j} &= G_{i+1,j} - 1/2\,(R_{i+2,j} + R_{i,j}) \\
Kr_{i,j-1} &= G_{i,j-1} - 1/2\,(R_{i,j-2} + R_{i,j}) \qquad (2) \\
Kr_{i,j+1} &= G_{i,j+1\,j} - 1/2\,(R_{i,j+2} + R_{i,j})
\end{aligned}$$

Where r,g,b are missing components and R,G,B are original values. The weights are then assigned to the four adjacent color difference values Kx , as defined In equation (2) ,The estimating $Kx_{(i,j)}$ calculated as follows

$$\begin{aligned}
K1 &= (Kr_{i-1,j}) / (1+\alpha_{i-1,j}) \\
K2 &= (Kr_{i+1,j}) / (1+\alpha_{i+1,j}) \\
K3 &= (Kr_{i,j-1}) / (1+\alpha_{i,j-1}) \qquad (3) \\
K4 &= (Kr_{i,j+1}) / (1+\alpha_{i,j+1})
\end{aligned}$$

$$Kx_{(i,j)} = (K1+K2+K3+K4) / ((1+\alpha_{i-1,j}) + (1+\alpha_{i+1,j}) + (1+\alpha_{i,j-1}) + (1+\alpha_{i,j+1})) \qquad (4)$$

Then the Missing Green pixel value in red CFA components is calculated as follows

$$\hat{G}_{i,j} = R_{i,j} + Kx_{(i,j)} \qquad (5)$$

Similarly The Green pixelvalue is calculated in Blue CFA components.

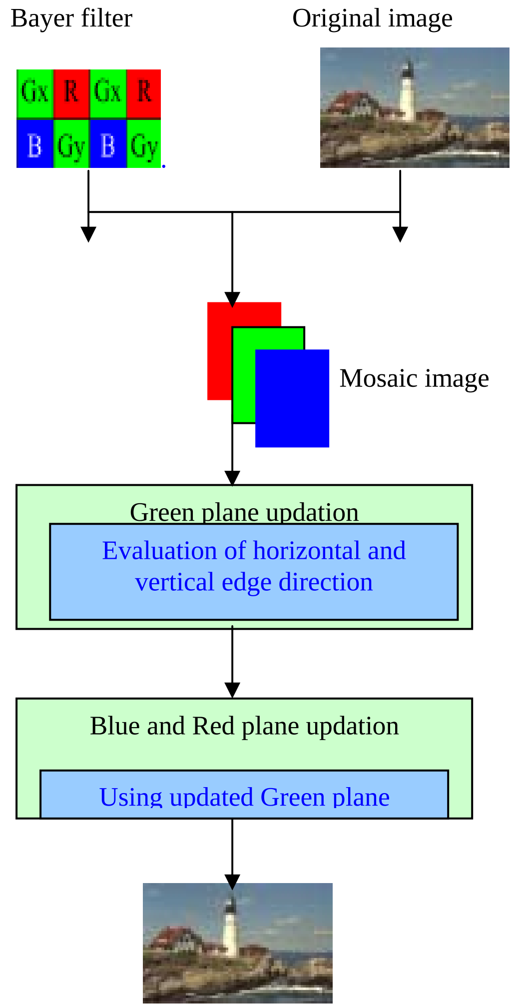


Fig.2 Demosaicing Algorithm

B Interpolating Missing Red Components at Green CFA sample position

After interpolating all missing green components of the image, the missing red and blue components at green CFA sampling positions are estimated. Figure 4.2(a) and 4.2 (b) shows the two possible cases where a green CFA sample is located at the center of a 5×5 block.

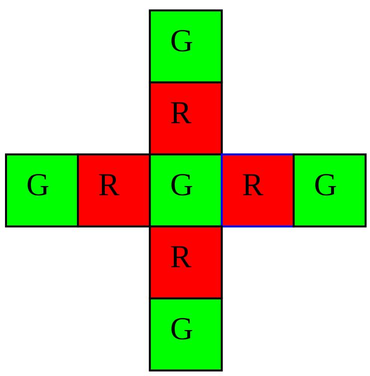
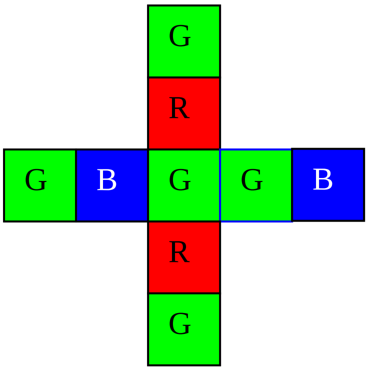


Figure 3.1 proposed CFA pattern having their centers at Green CFA samples

As for the case in Figure 4.3(a) ,the missing components of the centre are calculated as Follows

$\hat{R}_{i,j} = G_{i,j} + [R_{i-1,j} - \hat{G}_{i-1,j} + R_{i+1,j} - \hat{G}_{i+1,j}] / 2$



As for the case in Figure 4.3(b) ,the missing components of the centre are calculated as Follows

$\hat{R}_{i,j} = G_{i,j} + [R_{i,j-1} - \hat{G}_{i,j-1} + R_{i,j+1} - \hat{G}_{i,j+1}] / 2$

c Interpolating Blue Components at Red and Green CFA sample positions

Finally, the missing blue components at the red and green sampling positions are interpolated using the proposed algorithm .The blue components can be calculated based on the changing ratio of Green components in the same position . The missing blue sample, $b_{i,j}$, is interpolated by using the following algorithm

col = i mod 4

row = j mod 2

$B^{\wedge}_{i,j} = B_{(i-col+1,j-row+1)}$

$\Delta H_{i,j} = \hat{G}_{i,j} - \hat{G}_{(i-col+1,j-row+1)}$

$nWeightage = 1.0 * \Delta H_{i,j} / (0.001 + \hat{G}_{(i-col+1,j-row+1)})$

$B^{\wedge}_{i,j} = B^{\wedge}_{i,j} + nWeightage * B^{\wedge}_{i,j}$

C. Interpolating Missing Red and Blue Components at Green CFA Sampling Positions

After interpolating all missing green components of the image and enhancing the border, the missing red and blue components at green CFA sampling positions are estimated using the equation given below

$$R_{i,j} = G_{i,j} + \frac{R_{i,j-1} - G_{i,j-1} + R_{i,j+1} - G_{i,j+1}}{2} \qquad (8)$$

$$B_{i,j} = G_{i,j} + \frac{B_{i-1,j} - G_{i-1,j} + B_{i+1,j} - G_{i+1,j}}{2} \qquad (9)$$

D. Interpolating Missing Blue (Red) Components at Red (Blue) Sampling Positions

Finally, the missing blue (red) components at the red (blue) sampling positions are interpolated. The missing blue sample, $b_{i,j}$ is interpolated by

$$b_{i,j} = G_{i,j} + \frac{1}{4} \sum_{m=\pm1} \sum_{n=\pm1} (B_{i+m,j+n} - G_{i+m,j+n}). \qquad (10)$$

In a similar manner the missing red, $r_{i,j}$ is inter- polated by

$$r_{i,j} = G_{i,j} + \frac{1}{4} \sum_{m=\pm1} \sum_{n=\pm1} (R_{i+m,j+n} - G_{i+m,j+n}). \qquad (11)$$

The red and blue plane were further enhanced using some weighted function .For blue plane enhancement the formula is given by

$$\nabla g_{i,j-2} = \left| g_{i,j} - g_{i,j-2} \right|$$
$$\nabla g_{i,j+2} = \left| g_{i,j} - g_{i,j+2} \right| \quad (12)$$
$$b_{i,j} = g_{i,j} - \left(v_1 (g_{i,j-1} - b_{i,j-1}) + v_2 (g_{i,j+1} - b_{i,j+1}) \right), \nabla g_{i,j-2} < \nabla g_{i,j+2}$$
$$b_{i,j} = g_{i,j} - \left(v_2 (g_{i,j-1} - b_{i,j-1}) + v_1 (g_{i,j+1} - b_{i,j+1}) \right), \nabla g_{i,j-2} > \nabla g_{i,j+2}$$
$$b_{i,j} = g_{i,j} - \left(0.5 (g_{i,j-1} - b_{i,j-1}) + 0.5 (g_{i,j+1} - b_{i,j+1}) \right), \nabla g_{i,j-2} = \nabla g_{i,j+2}$$

where v_1 and v_2 are predefined weighted function, which is taken as 0.6 and 0.4 after examination of our simulation results. Using this relationship red plane can be enhanced by replacing r instead of b.

E. **Red and Blue Plane Border Enhancement**

After the interpolation it is observed that we get a poor border in the red and blue plane that is enhanced using the formula (7) replacing r and b instead of g.

F. Iteration

At this stage, the variance of each channel is evaluated. The variance of each plane is defined as follows

$$\mathrm{var}(X) = \left[\frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (X^{n+1}(i,j) - X^{n}(i,j) - m_x) \right]^2 \quad (13)$$

Where X represents R, G, or B, M and N separately represent the numbers of row or columns of the image, and m_x is the mean of the difference signal (X^{n+1},X^n),i.e.,

$$m_x = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(X^{n+1}(i,j) - X^{n}(i,j) \right) \quad (14)$$

For the G channel, G^0 stands for the outputs of the weighted edge interpolation, and G^1 for the outputs of the color difference interpolation, respectively. As to the R and B channels, R^1 and B^1 are the outputs of the color-difference interpolation, but both of R^0 and B^0 are set to be zero. In the iteration stage, the variance of each

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

TABLE I
PSNR Performance comparison of different Demosaicing Methods

IMAGE	Channel	POCS	PCSD	FSMU	Proposal Method
1	Red	28.4553	33.2122	35.2959	**35.3456**
	Green	32.9273	35.401	39.1361	**39.2313**
	Blue	28.142	33.0438	36.0125	35.9801
2	Red	31.4359	38.1016	34.4891	**37.2280**
	Green	33.3676	40.4894	40.914	**43.3307**
	Blue	31.2941	37.7678	38.9587	37.1122
3	Red	29.2315	35.311	36.3224	36.2321
	Green	33.259	37.5920	39.3066	39.0213
	Blue	29.2253	34.6224	35.4082	34.9087
4	Red	30.2407	31.6644	30.6963	**33.2211**
	Green	33.5075	33.9435	34.3265	**35.1234**
	Blue	30.3382	30.9675	30.7919	29.2213
5	Red	27.2408	29.6016	33.3178	32.3112
	Green	32.378	31.3597	35.5628	34.6781
	Blue	27.0262	29.3271	32.1294	31.3421
6	Red	31.1391	39.298	38.5834	**38.9786**
	Green	33.7569	41.5848	41.7205	**42.1237**
	Blue	30.8272	41.5848	38.1094	37.6789
7	Red	29.7384	34.8317	35.7577	**36.3456**
	Green	31.6029	36.833	38.9241	**39.1034**
	Blue	28.4014	34.1418	35.0944	**35.3456**
8	Red	26.4451	32.4751	33.7174	**34.4567**
	Green	28.0668	34.1149	36.1049	35.8902
	Blue	26.176	30.8661	31.8199	31.6789
9	Red	30.7306	38.2405	37.5531	**38.2345**
	Green	32.458	40.018	40.2358	**40.2456**
	Blue	30.5046	37.8739	37.3851	37.1234
10	Red	28.8603	33.3901	33.5461	33.1235
	Green	32.1476	35.0823	36.3242	**36.4567**
	Blue	30.5046	33.0352	33.677	32.7567
11	Red	27.7125	35.3082	35.9037	35.6789
	Green	30.7234	38.8958	39.2317	**40.1235**
	Blue	28.4945	36.301	36.3023	36.5674
12	Red	27.265	38.2058	37.5343	**38.4567**
	Green	28.8643	39.8856	40.1988	**41.5678**
	Blue	28.4662	36.4789	35.2987	35.0987

channel is compared with a universal threshold T3. If the variance of a channel is larger than T3, then update this signal by using the color difference interpolation. After this interpolation, a new variance of the channel is calculated. Finally, if all of the variances are no more than T3, then terminate the interpolation, or repeat the steps in iteration stage until the stopping condition is met or the maximum iteration number is reached.

III. SIMULATION RESULTS

Simulation was carried out to evaluate the performance of the proposed algorithm. The 12 digital color images shown in Fig. 2 were used to generate a set of testing images. The peak signal-to-noise ratio (PSNR)was used as a measure to quantify the performance of the demosaicing methods.In particular PSNR, it is defined as

$$CPSNR = 10\log_{10}\left(\frac{255^2}{CMSE}\right) \qquad (15)$$

Where

$$CMSE = (1/3HW)\sum_{i=r,g,b}\sum_{y=1}^{H}\sum_{x=1}^{W}((I_0(x,y,i) - I_r(x,y,i))^2$$

I_o and I_r represent, respectively, the original and reconstructed images of size H × W each.

Table I shows the performance of various algorithms .the proposed algorithm provided the best performance among the evaluated algorithm. The proposed algorithm is developed based on the fact that the interpolation direction for each missing green sample is critical to the final demosaicing result.

IV. CONCLUSION

In this paper, a new iterative demosaicing algorithm with enhanced border and edge was presented. It makes use of the color difference variance of the pixels located along the horizontal axis and that along the vertical axis in a local region to estimate the interpolation direction for interpolating the missing green samples. The high-performance arises from the introduction of a weighted edge interpolation and the well designed stopping strategy. With them, the proposed algorithm has a good initial condition and can terminate iteration early. Simulation results show that the proposed algorithm is able to produce a subjectively and objectively better demosaicing results as compared with a number of advanced algorithms.

References

[1] C.-Y. Su, "Effective Iterative Demosaicing Using Weighted Edge and Color Difference Interpolations," submitted to IEEE Trans. Consumer Electronics., March 2008

[2] K.H. Chung and Y.H. Chan, Color Demosaicing Using Variance of Color Differences, IEEE Trans. on Image Processing, Vol. 15, No.10, pp.2944-2955, Oct 2006.

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

[4] X. Li, “Demosaicing by successive approximation,” IEEE Trans. Image Process., vol. 14, no. 3, pp. 370-379, March 2005.

[5] W. Lu and Y.-P. Tan, “Color filter array demosaicing: New method and performance measures,” IEEE Trans. Image Process., vol. 12, no. 10, pp. 1194-1210, Oct. 2003

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

[7] K.Hirakawa and T.W.Parks,”Adaptive homogeneity – directed demosaicing algorithm,” IEEE Trans. on Image Process, vol. 141, no. 39, pp. 360-369, Mar. 2005.

[8] D. Alleysson, S. Süsstrunk, and J. Hérault, “Linear demosaicing inspired by the human visual system,” *IEEE Trans. Image Process*., vol. 14, no. 4, pp. 439-449, April 2005.

[9] A. Heinrich, G. de Haan, and C. N. Cordes, "A Novel Performance Measure for Picture Rate Conversion Methods," *Consumer Electronics,2008. ICCE 2008. Digest of Technical Papers. International Conference on* , vol., no., pp.1-2, 9-13 Jan. 2008