Directionally Weighted Color Interpolation for Digital Cameras

Hung-An Chang[1] *and Homer Chen*[1,2]

[1]Department of Electrical Engineering, [2]Graduate Institute of Communication Engineering
National Taiwan University, Taipei, Taiwan, R.O.C.
E-mail: hungana@ms56.hinet.net, homer@cc.ee.ntu.edu.tw

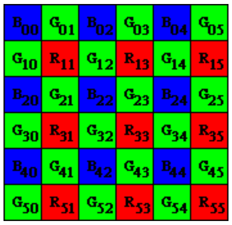
Abstract—**Demosaicing is a color interpolation process that converts a raw image generated by a color filter array to a full color image by estimating the missing color components of each pixel from its neighbors. Most demosaicing approaches often introduce false colors or blur edges to areas of the image where there are dense edges. In this paper, we present a directionally weighted color interpolation method to address this problem. We consider finer edge orientations to improve the accuracy of edge indicators and exploit the inter-channel correlation of R, G, and B color planes in the interpolation process to reduce the undesirable artifacts and hence improve the perceptual image quality. Experimental results are shown to demonstrate the strengths of the proposed method.**

I. Introduction

Most commercially available digital cameras use a color filter array (CFA) to capture images with a single CCD sensor array, resulting in a sub-sampled raw image with a single *R*, *G*, or *B* component for each pixel of the image. To reconstruct a full color image from the raw sensory data, a color interpolation scheme is required to estimate the two missing color components of each pixel. In this paper, we consider the interpolation for Bayer color arrays shown in Fig. 1.

The false colors and blur edges, often happening in image areas where there are dense edges, are due to the aliasing effects caused by averaging (low-pass filtering) image samples across the edges in the interpolation process. Many methods have been proposed [1]-[8] to solve the problem. These methods can be classified into two categories: non-adaptive interpolation and edge-directed adaptive interpolation.

For non-adaptive methods, the interpolation is carried out uniformly across the image. These methods reduce the aliasing by performing the interpolation in a different color space rather than the original *RGB* space. The smooth hue approach [1] and the color difference approach [4] belong to this category. Although such methods are relatively simple to implement, their effectiveness is limited, for the lack of adaptivity near edge regions where aliasing is likely to occur.



B_{00}	G_{01}	B_{02}	G_{03}	B_{04}	G_{05}
G_{10}	R_{11}	G_{12}	R_{13}	G_{14}	R_{15}
B_{20}	G_{21}	B_{22}	G_{23}	B_{24}	G_{25}
G_{30}	R_{31}	G_{32}	R_{33}	G_{34}	R_{35}
B_{40}	G_{41}	B_{42}	G_{43}	B_{44}	G_{45}
G_{50}	R_{51}	G_{52}	R_{53}	G_{54}	R_{55}

Figure 1. Bayer pattern.

The adaptive methods, on the other hand, adjust the weights of samples in the interpolation process by utilizing the edge information gathered by edge indicators. If an edge indicator senses a possible color intensity change across the direction of an edge, a small weight is assigned to the sample along that edge direction. Therefore, the effectiveness of an edge-adaptive method highly depends on its edge sensing capability.

In this paper, we propose a directionally weighted color interpolation method. By carefully designing the edge indicator for each edge direction, a good edge sensing performance is achieved. In our method, the color difference technique is also employed to exploit the inter-channel correlation between color planes. The details of this method are described in Sec. II. Experiments and performance comparisons are presented in Sec. III, followed by a conclusion in Sec. IV.

II. Directionally Weighted color onterpolation

A. *Fine-Resolution Edge Indicators*

To increase the edge sensitivity, we consider eight more samples in addition to the four commonly used samples for interpolation. This is illustrated in Fig. 2. Take the interpolation of the missing *G* at B_{44} as an example. The four nearby samples G_{43}, G_{34}, G_{45}, and G_{54} are commonly used by most edge-directed algorithms. The additional 8 samples G_{32}, G_{23}, G_{25}, G_{36}, G_{56}, G_{65}, G_{63}, and G_{52} considered in our approach are located one knight step away from B_{44}. Here, a

0-7803-8834-8/05/$20.00 ©2005 IEEE.

knight step is a combination of one vertical step plus two horizontal steps, or vise versa, on the Bayer color array. For notational convenience, we number the directions from 1 to 12 as shown in Fig. 2. For a given sample, the nearest sample in the same color plane is either 2 or $2\sqrt{5}$ pixels away in each edge direction. For example, the nearest sample of B_{44} in direction 3 is B_{46}, which is 2 pixels away from B_{44}. Likewise, the nearest sample of B_{44} in direction 8 is B_{28}, which is $2\sqrt{5}$ pixels away.

In this multi-directional configuration, the difference of nearby samples (referred to as sample difference) along a direction is not a fair measure of the edge strength, because the nearest samples are not equally distanced in all directions. A proper adjustment on the sample difference is required to put the measure of edge strength along all edge directions on the same footing. There are two possible approaches.

The first approach is called linear adjustment. In this approach, a sample difference is normalized by the distance between the two samples based on which the sample difference is computed. Edge indicators of this kind are also referred to as directional derivatives [3].

The second approach, called stochastic adjustment, is based on the assumption that the image sample X can be modeled as a locally stationary Gaussian process with mean μ_X and standard deviation σ_X. Under this assumption, $\Omega(d)$, the correlation of two samples with distance d between them, can be approximated by $exp(-d^2/\rho^2)$ [2], where ρ is an image-dependent parameter, and the sample difference can be modeled as a Gaussian random variable with zero mean and standard deviation σ_d.

$$\begin{aligned}\sigma_d &= \sqrt{E[(X(0)-X(d))^2]}\\ &= \sqrt{E[(X'(0)-X'(d))^2]}\\ &= \sqrt{E[(X'(0))^2]+E[(X'(d))^2]-2E[X'(0)X'(d)]}\\ &= \sigma_X\sqrt{2(1-\Omega(d))},\end{aligned} \qquad (1)$$

where $X(0)$ and $X(d)$, respectively, denote the current sample and a sample that is d pixels away from $X(0)$, $X'(0)=X(0)-\mu_X$, and $X'(d)=X(d)-\mu_X$. Normalizing the sample differences by σ_d is equivalent to multiplying the sample differences in directions 5-12 by a factor κ_n while keeping the sample differences in directions 1-4 unchanged.

$$\begin{aligned}\kappa_n &= \sigma_X\sqrt{2(1-\Omega(2))}\Big/(\sigma_X\sqrt{2(1-\Omega(2\sqrt{5}))})\\ &\approx\sqrt{(1-\Omega(2))\Big/(1-(\Omega(2)^5))}, \quad 5\le n\le 12\end{aligned} \qquad (2)$$

where the value of κ_n is computed based on the approximated $\Omega(d)$. κ_n is around 0.5 as long as $\Omega(2)$ is high enough. For example, κ_n=0.519 when $\Omega(2)$= 0.85, κ_n=0.5 when $\Omega(2)$=0.9, and κ_n=0.472 when $\Omega(2)$=0.95.

For the two approaches, the edge indicator I_n for direction n can be obtained by:

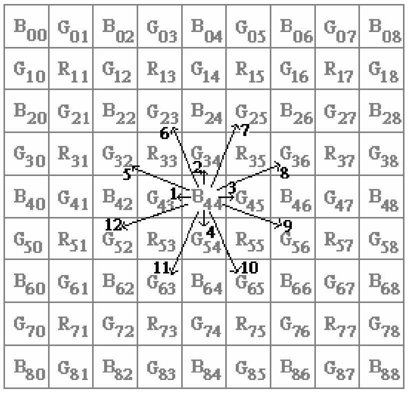

B00 G01 B02 G03 B04 G05 B06 G07 B08
G10 R11 G12 R13 G14 R15 G16 R17 G18
B20 G21 B22 G23 B24 G25 B26 G27 B28
G30 R31 G32 R33 G34 R35 G36 R37 G38
B40 G41 B42 G43 B44 G45 B46 G47 B48
G50 R51 G52 R53 G54 R55 G56 R57 G58
B60 G61 B62 G63 B64 G65 B66 G67 B68
G70 R71 G72 R73 G74 R75 G76 R77 G78
B80 G81 B82 G83 B84 G85 B86 G87 B88
1
2
3
4
5
6
7
8
9
10
11
12

Figure 2. Directions of interpolation.

Table 1. POSITIONS OF NEARBY SAMPLES IN STEPS 1 AND 3.

n	v_n	h_n	n	v_n	h_n
1	0	-1	2	-1	0
3	0	+1	4	+1	0
5	-1	-2	6	-2	-1
7	-2	+1	8	-1	+2
9	+1	+2	10	+2	+1
11	+2	-1	12	+1	-2

$$\begin{aligned}I_n(i,j) = \kappa_n(&abs(P(i+v_n, j+h_n)-P(i-v_n, j-h_n))+\\ &abs(P(i+2v_n, j+2h_n)-P(i,j)))\end{aligned} \qquad (3)$$

where $P(i,j)$ denotes the sample at the position (i, j) and h_n and v_n, listed in Table 1, denote the horizontal and vertical positions, respectively, of a nearest sample relative to the sample to be interpolated. In our experiments, we set κ_n to the following values:

$$\kappa_n = \begin{cases} 1, & 1\le n\le 4, \text{ for linear and stochastic adjustment}\\ 1\big/\sqrt{5}, & 5\le n\le 12, \text{ for linear adjustment}\\ 0.5, & 5\le n\le 12, \text{ for stochastic adjustment}\end{cases} \qquad (4)$$

Experimental results of these two approaches are compared in Sec. 3.

B. Directionally Weighted Interpolation

Because of high correlation between the R, G, and B channels [4], the color difference values (G-R, G, G-B) are relatively smoother than the (R, G, B) values, making the aliasing effects less pronounced in the color difference space. Like some previous methods, our interpolation method is conducted in the color difference space.

Let $w_n(i,j)$, be the weight for direction n when interpolating a missing color at (i,j).

$$w_n(i,j) = \left(\frac{1}{1+I_n(i,j)}\right)\Big/\sum_{n=1}^{12}\frac{1}{1+I_n(i,j)}, \qquad (5)$$

where $I_n(i,j)$ is obtained from (3).

Our interpolation algorithm consists of four steps:

Step 1. Interpolate the missing green value of blue/red samples.

The missing green value *G(i,j)* of a blue sample *B(i,j)* is determined by:

$$G(i,j) = B(i,j) + \sum_{n=1}^{12} w_n(i,j) * K_{b,n}(i+v_n, j+h_n) \tag{6}$$

where

$$K_{b,n}(i+v_n, j+h_n) = G(i+v_n, j+h_n) - B(i+v_n, j+h_n) \tag{7}$$

is the color difference along direction *n*. The rule of calculating the color difference is as follows. If the value of *B* (or *G*) is available at the target location in the Bayer array, use it directly; otherwise, check to see if it has been interpolated previously. Use the interpolated value if it exists. If it does not, take the average of the two adjacent samples of the desired color in the Bayer array. Refer to Fig. 2 for an example. To interpolate the missing green value at *(4,4)*, $K_{b,6}(2,3)$ is needed and is obtained by calculating $G_{2,3} - (B_{2,2}+B_{2,4})/2$. Note again that the color difference, as opposed to green difference [3], is adopted here to enhance the accuracy of the interpolation. The missing green value of a red sample can be obtained in a way similar to (6) and (7).

Step 2. Interpolate the missing red/blue values of blue/red samples.

In this step, we consider only the four nearest samples of the same color in the diagonal directions because there are no similar samples in the other 8 directions (Fig. 2). The vertical and horizontal positions of these four samples (indexed from *n*=1 to *n*=4) relative to the sample to be interpolated are listed in Table 2. Let $I_n^{'}(i,j)$ be the edge indicator for direction *n*.

$$\begin{aligned} I_n^{'}(i,j) = {} & abs(P(i+v_n^{'}, j+h_n^{'}) - P(i-v_n^{'}, j-h_n^{'})) + \\ & abs(P(i+2v_n^{'}, j+2h_n^{'}) - P(i,j). \end{aligned} \tag{8}$$

Denote the weight for direction *n* when interpolating a missing color at *(i,j)* by $w_n^{'}(i,j)$,

$$w_n^{'}(i,j) = \left(\frac{1}{1+I_n^{'}(i,j)}\right) \Bigg/ \sum_{n=1}^{4} \frac{1}{1+I_n^{'}(i,j)}. \tag{9}$$

The missing red value of a blue sample *B*(*i,j*) is obtained by:

$$R(i,j) = G(i,j) - \sum_{n=1}^{4} w_n^{'}(i,j) * K_{r,n}(i+v_n^{'}, j+h_n^{'}), \tag{10}$$

where

$$K_{r,n}(i+v_n^{'}, j+h_n^{'}) = G(i+v_n^{'}, j+h_n^{'}) - R(i+v_n^{'}, j+h_n^{'}). \tag{11}$$

is the color difference along direction *n*.

The interpolation of missing blue values of red samples is similar to (10) and (11).

TABLE 2. POSITIONS OF NEARBY SAMPLES IN STEP 2.

n	v'_n	h'_n	*n*	v'_n	h'_n
1	-1	-1	2	-1	+1
3	+1	+1	4	+1	-1

Step 3. Interpolate missing red/blue values of green samples.

The missing red value of a green sample *G*(*i,j*) is obtained by:

$$R(i,j) = G(i,j) - \sum_{n=1}^{12} w_n(i,j) * K_{r,n}(i+v_n, j+h_n), \tag{12}$$

where $w_n(i,j)$ is computed by (5) . The interpolation of the remaining missing blue values is similar to (12).

Step 4. Adjust the estimated green values of red/blue samples.

In step 1 the blue (red) values required to calculate the color differences are computed by averaging the values of adjacent samples. This average operation, however, will cause some aliasing. To reduce this aliasing, we re-compute the green values interpolated in step 1 by using the blue (red) values obtained in step 3 when calculating the color differences as in (7).

III. EXPERIMENTS AND COMPARISONS

The directionally weighted interpolation method is designed for reducing the artifacts around dense edges. To evaluate its effectiveness, the performance of the proposed method was compared with that of four existing color interpolation methods: the bilinear method, the color difference based method [4], the gradient-based method [1], and the C2D2 method [3].

The bilinear interpolation method simply averages the most nearby samples to interpolate missing color samples, and introduces serious artifacts near the edges. The color difference based method [4] considers same set of nearby samples as the bilinear interpolation, but utilizes advantages of the color differences. The gradient based method [1] is a hard decision, edge-directed method. It compares the horizontal and vertical gradients and chooses samples in the direction with smaller gradient for interpolation. The C2D2 (Color Correlations and Directional Derivatives) [3] method is an edge-directed method considering 8 directions for interpolation. The relative weights for these directions are calculated by the directional derivatives. Besides of these four methods, the performances of two adjustment approaches described in Sec. II are also compared.

The results of two test images are shown in Fig. 3-4 respectively. Fig. 3 is the down-right portion of a building, which contains very fine textures below each window. The image in Fig. 4 is cropped from the airplane image of Kodak image database [3]. The PSNR values of resulting images in Fig. 3-4 are listed in Table 3 and Table 4 respectively. Note that only the pixels shown in Fig. 3-4 are considered for the computation of PSNR to evaluate the performances of these methods near the edge regions. According to the experiment results, the artifacts are significantly reduced by our method. Besides, in terms of the PSNR results, the stochastic adjustment approach outperforms the linear adjustment approach. The results of the full images in Fig. 3-4 as well as results of other benchmark images are available on http://www2.ee.ntu.edu.tw/~b9901049/demosaicing.html.

IV. CONCLUSION

In this paper, we have described a directionally weighted color interpolation method for digital cameras using a color filter array to acquire the image. A general scheme for extracting edge information at finer orientation than previous approaches was also presented. By going for finer edge directions and exploiting inter-channel correlation between color planes, our method is able to achieve a better edge-sensing capability. The comparison results demonstrate the robustness of our method in dealing with densely spaced edges.

REFERENCES

[1] B. Eamanath, W.E. Synder, and G. L. Bilbro, "Demosiacking methods for Bayer color array," Journal of Electronic Imaging 11(3), pp. 306-315, Jul. 2002

[2] X. Li and M.T. Orchard, "New edge-directed interpolation," IEEE Trans. on Image Processing, vol.10, no. 10, pp. 1521-1527, Oct. 2001

[3] N. Kehtarnavaz, H. Oh, and Y. Yoo, "Color filter array interpolation using correlations and directional derivatives," Journal of Electronic Imaging 12(4), pp. 621-632, Oct. 2003

[4] S.C. Pei and I. K. Tam, "Effective color interpolation in CCD color filter arrays using signal correlation," IEEE Trans. on Circuit and System for Viedo Technology, vol. 13, no. 6, Jun. 2003.

[5] R. Lukac, K.N. Plataniotis, D. Hatzinakos, and M. Aleksic, "A novel cost effective demosaicing approach," IEEE Trans. on Consumer Electronics, vol.50, no. 1, Feb. 2004.

[6] B. S Hur and M. G. Kang , "High definition color interpolation scheme for progressive scan CCD image sensor," IEEE Trans. on Consumer Electronics, vol. 47, no. 1 Feb. 2001.

[7] B. K. Gunturk, Y. Altunbasak, R. M. Mersereau, "Color plane interpolation using alternating projections," IEEE Trans. on Image Processing, vol. 11, no. 9, Sept. 2002.

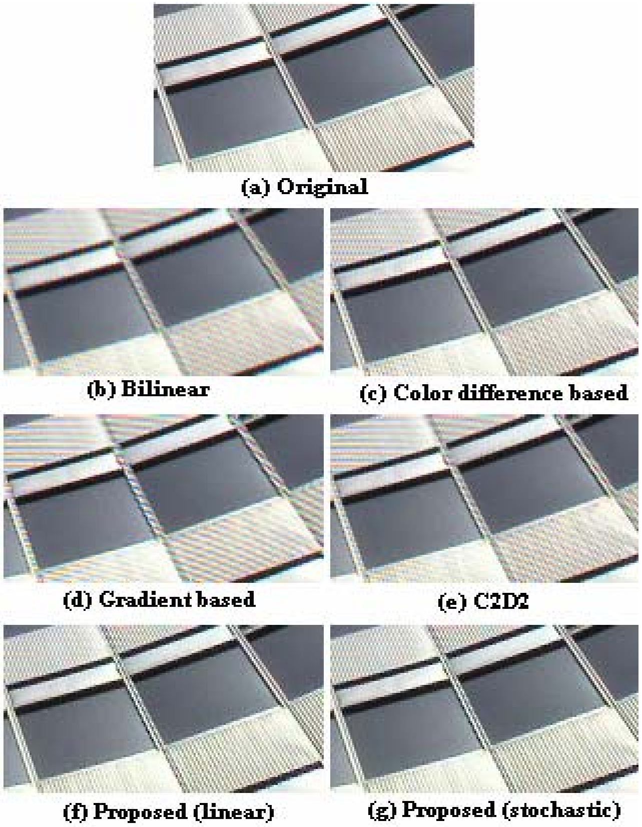
[8] C.L. Hsu, *Demosaicking algorithm with noise removal based on adaptive spatial filter*, Master Thesis, GICE, National Taiwan University

TABLE 3. PSNR COMPARISON FOR FIGURE 3.

	PSNR_G	PSNR_R	PSNR_B
Bilinear	19.63	18.94	19.07
Color difference based	24.50	24.23	25.12
Gradient based	18.05	19.36	19.60
C2D2	23.39	23.48	24.51
Proposed (linear)	27.38	24.47	26.45
Proposed (stochastic)	27.49	24.58	26.59

TABLE 4. PSNR COMPARISON FOR FIGURE 4.

	PSNR_G	PSNR_R	PSNR_B
Bilinear	23.15	21.50	21.61
Color difference based	29.85	28.51	29.20
Gradient based	24.51	26.57	26.27
C2D2	30.32	31.45	31.00
Proposed (linear)	34.64	31.96	31.98
Proposed (stochastic)	34.77	32.08	32.08



(a) Original

(b) Bilinear

(c) Color difference based

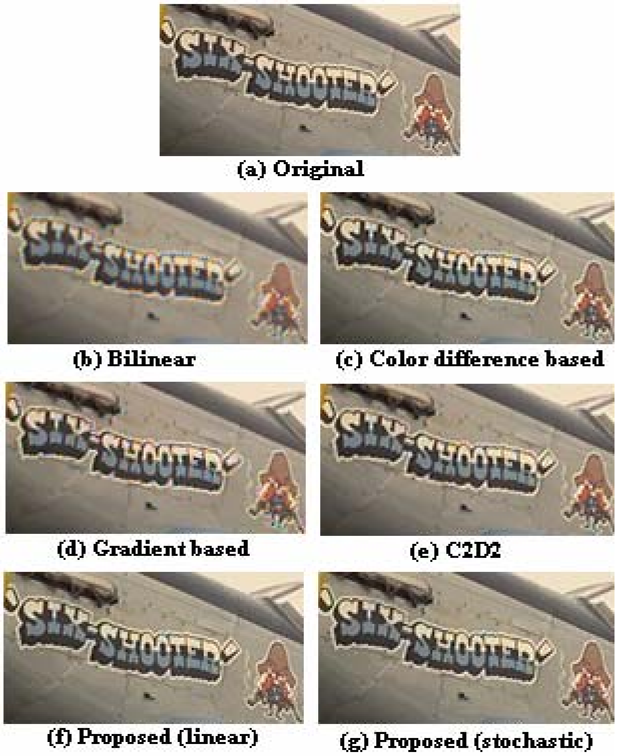
(d) Gradient based

(e) C2D2

(f) Proposed (linear)

(g) Proposed (stochastic)

Figure 3. Comparison of the methods for the building image.



(a) Original

(b) Bilinear

(c) Color difference based

(d) Gradient based

(e) C2D2

(f) Proposed (linear)

(g) Proposed (stochastic)

Figure 4. Comparison of the methods for the airplane images. Note the aliasing effects near the edges of the logo.