Color Filter Array Interpolation for Edge Strength Filters

L.Naveen[#1], B.Shobanbabu[*2], M.Tech (Ph. d)

[1]*Sri Venkateswara College of Engineering & Technology, JNTUA,*
R.V.S.Nagar, Chittoor

[2]*Associate Professor, Dept. ECE, Sri Venkateswara College of Engineering & Technology, JNTUA,*
R.V.S.Nagar, Chittoor

***Abstract*— Basically we use beam splitters to capture data or image. Using beam splitters is very expensive, so inorder to overcome this we go for an alternative technique called color filters arrays. As a result of this, only one of the required three color samples becomes available at each pixel location and the other two need to be interpolated. This process is called Color Filter Array (CFA) interpolation or demosaicing. Many demosaicing algorithms have been introduced over the years to improve subjective and objective interpolation quality. We propose an orientation-free edge strength filter and apply it to the demosaicing problem. Edge strength filter output is utilized both to improve the initial green channel interpolation and to apply the constant color difference rule adaptively. This simple edge directed method yields visually pleasing results with high CPSNR.**

***Keywords*— Color filter array (CFA) interpolation, demosaicing, edge directed interpolation, orientation-free edge filter.**

I. Introduction

Color images require multiple data samples for each pixel as opposed to grayscale images for which a pixel is represented by only one data sample. For the RGB image format, these data samples represent red, green and blue channels . A typical digital camera captures only one of these channels at each pixel location and the other two need to be estimated to generate the complete color information. This process is called color filter array (CFA) interpolation or demosaicing. Although many different CFA patterns have been proposed, the most prevalent one is the Bayer pattern shown in Fig. 1 [1]. As an important step in image processing pipeline of digital cameras, demosaicing has been an area of interest in both academia and industry. The simplest approach to the demosaicing problem is to treat color channels seperately and fill in missing pixels in each channel using a spatially invariant interpolation method such as bilinear or bicubic interpolation. While such an approachworks fine in homogenous areas, it leads to color artifacts and lower resolution in regions with texture and edge structures. Obtaining better demosaicing performance is possible by exploiting the correlation between the color channels. Spectral correlation can be modeled by either constant color ratio rule [2], [3] or constant color difference rule [4], [5]. The basic assumption is that color ratio/difference is constant over a local distance inside a given object. This assumption is likely to break apart across boundaries, hence many demosaicing algorithms try to utilize it adaptively in one way or another. Since the Bayer CFA pattern has twice as many green channel samples as red and blue ones, green channel suffers less from aliasing and is the natural choice as the starting point of the CFA interpolation process. In [6], Glotzbach et al. proposed improving red and blue channel interpolation by adding high frequency components extracted from green channel to red and blue channels.

In another frequency-domain approach, Gunturk et al. [7] used an alternating projections scheme based on strong inter-channel correlation in high frequency subbands. Although the main objective is to refine red and blue channels iteratively, the same approach can also improve green channel interpolation beforehand which in turn yields better red and blue channel results.

A more recent method [8] makes several observations about color channel frequencies and suggests that filtering the CFA image as a whole instead of individual color channels should preserve high frequency information better. To estimate luminance, the method proposes a fixed 5-by-5 filter at green pixel locations and an adaptive filter for red and blue pixel locations. The estimated full resolution luminance is then used to complete missing the chrominance information. Edge-directed green channel interpolation has been proposed early on with various direction decision rules [4], [5], [9], [10]. The method outlined in [4] is particularly noteworthy because it proposed using derivatives of chrominance samples in initial green channel interpolation.

Several subsequent demosaicing algorithms made use of this idea. Authors of [11] proposed using variance of color differences as a decision rule while Zhang et al. [12] proposed making a soft decision to improve the interpolation performance of the original method [4]. In this method [12], color differences along horizontal and vertical directions are treated as noisy observations of the target pixel color difference and they are combined optimally using the linear minimum mean square error estimation (LMMSE) framework. Paliy et al. [13] further improved directional filtering proposed in [12] by introducing scale adaptive filtering based on linear polynomial approximation (LPA). Several methods proposed performing interpolation in both horizontal and vertical directions and making a posteriori decision based on some criteria. Hirakawa et al. [15] compared local homogeneity of horizontal and vertical interpolation results and Menon et al. [16] used color gradients over a local window to make the direction decision. The rest of the paper is organized as follows. Section II describes the proposed CFA interpolation algorithm. Section III presents experimental results, and Section IV reports the conclusions.

II. PROPOSED ALGORITHM

The basis of the proposed algorithm is the observation that the constant color difference assumption tends to fail across edges. If one can effectively utilize edge information to avoid averaging non-correlated color differences, demosaicing performance could increase dramatically. The question at this point is, how the edge information can be expressed meaningfully at the pixel level so that it is useful enough to improve demosaicing performance. Edge detection filters such as Sobel and Canny can tell whether an edge structure is present at a given pixel. However, they do not provide any information about the sharpness of luminance transition at that particular pixel. We propose an edge strength filter that provides local, orientation free luminance transition information. The filter has a 3 by 3 support size. Given a grayscale input image, it could be formulated as:

$$S_{p6} = \frac{|P_1 - P_{11}|}{2} + \frac{|P_3 - P_9|}{2} + |P_2 - P_{10}| + |P_5 - P_7| \qquad (1)$$

Where Sp6stands for the edge strength at pixel location P6.

By applying the filter to all available pixels, we get the edge strength map of the input image. Note that, although the filter result for a single pixel does not provide any edge direction information, the relationship between neighboring pixel results yields the edge orientation in that neighborhood. The proposed filter is very useful for finding edges in a grayscale image. However, a mosaicked image only has one of the three color channels available for every pixel location and it certainly does not have complete luminance information at any pixel. That is why, the edge strength filter can only be applied to a mosaicked image by making an approximation. Instead of trying to estimate luminance information and taking estimated luminance differences of neighboring pixel pairs, we take the difference in terms of the available color channel for each pixel pair. For instance, for the red center pixel case the diagonal differences will come from the blue channel and the rest from the green channel:

$$S_{R_{10}} = \frac{|B_5 - B_{15}|}{2} + \frac{|B_7 - B_{13}|}{2} + |G_6 - G_{14}| + |G_9 - G_{11}| \qquad (2)$$

The edge strength for green and blue pixels will be calculated in the same way. The edge strength map obtained from the mosaicked input image will help us both in initial green channel interpolation stage and in subsequent green channel update.

A. Green Channel Interpolation

We propose making a hard decision based on the edge strength filter described above. For this purpose, every green pixel to be interpolated (red or blue pixel in the mosaicked image) is marked either horizontal or vertical by comparing the edge strength differences along each direction on a local window. For a window size of 5 by 5, horizontal and vertical difference costs can be formulated as follows:

$$H_{i,j} = \sum_{m=-2}^{2}\left(\sum_{n=-2}^{1}\left(S_{i+m,j+n} - S_{i+m,j+n+1}\right)\right) \qquad (3)$$

$$V_{i,j} = \sum_{m=-2}^{1}\left(\sum_{n=-2}^{2}\left(S_{i+m,j+n} - S_{i+m+1,j+n}\right)\right) \qquad (4)$$

Where $S_{i,j}$ is the edge strength filter output at pixel location $S_{i,j}$ and $H_{i,j}$ and $V_{i,j}$ represent the total horizontal and vertical costs, respectively.

The target pixel will be labeled horizontal if horizontal cost is less than vertical and vice versa. The rationale behind this decision scheme is that if there happens to be a horizontal edge in a given neighborhood, then the edge strength differences between vertical neighbors will vary more than those of horizontal neighbors. After all the pixels are labeled, the robustness of the direction decision can be improved by relabeling them based on the directions of their neighbors. For instance, considering the closest 8 neighbors of a target pixel and the pixel itself, it will be labelled horizontal only if more than 4 of those 9 pixels are initially labelled horizontal. Based on the final direction label, green channel is interpolated as follows:

$$\bar{G}_{i,j} = \begin{cases} B_{i,j} + \dfrac{\bar{G}^H_{i,j} - B_{i,j}}{2} + \dfrac{G_{i,j-1} - \bar{B}^H_{i,j-1}}{4} + \dfrac{G_{i,j+1} - \bar{B}^H_{i,j+1}}{4} & \text{if Horizontal} \\ B_{i,j} + \dfrac{\bar{G}^V_{i,j} - B_{i,j}}{2} + \dfrac{G_{i-1,j} - \bar{B}^V_{i-1,j}}{4} + \dfrac{G_{i+1,j} - \bar{B}^V_{i+1,j}}{4} & \text{if Vertical} \end{cases} \qquad (4)$$

where directional estimations are calculated by:

$$\bar{G}^{H}_{i,j} = \frac{G_{i,j-1}+G_{i,j+1}}{2} + \frac{2*B_{i,j}-B_{i,j-2}-B_{i,j+2}}{4} \quad (5)$$

$$\bar{G}^{V}_{i,j} = \frac{G_{i-1,j}+G_{i+1,j}}{2} + \frac{2*B_{i,j}-B_{i-2,j}-B_{i+2,j}}{4} \quad (6)$$

$$\bar{B}^{H}_{i,j} = \frac{B_{i,j-1}+B_{i,j+1}}{2} + \frac{2*G_{i,j}-G_{i,j-2}-G_{i,j+2}}{4} \quad (7)$$

$$\bar{B}^{H}_{i,j} = \frac{B_{i-1,j}+B_{i+1,j}}{2} + \frac{2*G_{i,j}-G_{i-2,j}-G_{i+2,j}}{4} \quad (8)$$

Green channel estimation for red pixel locations is performed simply by replacing R's with B's in the equations above.

B.Green Channel Update

The second step of the proposed algorithm is updating the green channel. We make use of the constant color difference assumption combined with edge strength filter to improve the initial green channel interpolation while avoiding averaging across edge structures. For every green pixel to be updated, the closest four neighbors with available color difference estimates are considered.

We expect the edge strength difference between two pixels to be large across edges. That is why the weight for each neighbor is inversely correlated with the total absolute edge strength difference in its direction. In other words, a neighbor will contribute less to the update result if there happens to be a strong edge between the target pixel and itself. Assuming we are updating the green channel value at a blue pixel:

$$D_1 = |S_{i,j} - S_{i-1,j}| + |S_{i-1,j} - S_{i-2,j}| + |S_{i-2,j} - S_{i-3,j}| + C_1 \quad (9)$$

$$D_2 = |S_{i,j} - S_{i,j-1}| + |S_{i,j-1} - S_{i,j-2}| + |S_{i,j-2} - S_{i,j-3}| + C_1 \quad (10)$$

$$D_3 = |S_{i,j} - S_{i,j+1}| + |S_{i,j+1} - S_{i,j+2}| + |S_{i,j+2} - S_{i,j+3}| + C_1 \quad (11)$$

$$D_4 = |S_{i,j} - S_{i+1,j}| + |S_{i+1,j} - S_{i+2,j}| + |S_{i+2,j} - S_{i+3,j}| + C_1 \quad (12)$$

$$M_1 = D_2 * D_3 * D_4 \qquad M_2 = D_1 * D_3 * D_4$$

$$M_3 = D_1 * D_2 * D_4 \qquad M_4 = D_1 * D_2 * D_3$$

$$\hat{G}_{i,j} = B_{i,j} + W * (\tilde{G}_{i,j} - B_{i,j}) + (1 - W)$$

$$*[\frac{M_1}{M_{Total}}(\tilde{G}_{i-2,j} - B_{i-2,j}) + \frac{M_2}{M_{Total}}(\tilde{G}_{i,j-2} - B_{i,j-2}) +$$

$$\frac{M_3}{M_{Total}}(\tilde{G}_{i,j+2} - B_{i,j+2}) + \frac{M_4}{M_{Total}}(\tilde{G}_{i+2,j} - B_{i+2,j})$$

$$M_{Total} = M_1 + M_2 + M_3 + M_4 \quad (13)$$

Again, green channel values at red pixel locations are updated in the same way by replacing B's with R's in the equations above. stands for updated green channel result while is the initial green channel interpolation. C_1 is a nonzero constant to avoid zero denominators. is the weight for the initial color difference estimation and is the neighbors' contribution to the green channel update. Updating green channel reduces color artifacts and improves PSNR. However zipper artifacts become more prominent as the number of updates increase. Experiments on test images suggest that one or two green channel updates are adequate. The performance of green channel update can be improved further by making adaptive for each pixel by checking the total absolute difference between the closest known green pixels. The idea is that green channel update should be more aggressive if there happens to be a lot of difference between known green pixels in that neighborhood because initial interpolation is more likely to fail in such areas.

C. Red and Blue Channel Interpolation

Once the green channel interpolation is finalized, we fill in red and blue channels using constant color difference assumption. For red channel interpolation at blue pixels and blue channel interpolation at red pixels, diagonal neighbors are used adaptively based on green channel gradients in both directions:

$$M_1 = |\hat{G}_{i-2,j-2} - \hat{G}_{i,j}| + |\hat{G}_{i-1,j-1} - \hat{G}_{i+1,j+1}| + |\hat{G}_{i,j} - \hat{G}_{i+2,j+2}|$$

$$M_2 = |\hat{G}_{i-2,j+2} - \hat{G}_{i,j}| + |\hat{G}_{i-1,j+1} - \hat{G}_{i+1,j-1}| + |\hat{G}_{i,j} - \hat{G}_{i+2,j-2}| \quad (14)$$

If coordinate is a red pixel location, blue channel estimation is calculated by:

$$\hat{B}_{i,j} = \hat{G}_{i,j} - \frac{M_2 * (\hat{G}_{i-1,j-1} - B_{i-1,j-1} + \hat{G}_{i+1,j+1} - B_{i+1,j+1})}{2 * (M_1 + M_2)} - \frac{M_1 * (\hat{G}_{i-1,j+1} - B_{i-1,j+1} + * (\hat{G}_{i+1,j-1} - B_{i+1,j-1}}{2 * (M_1 + M_2)} \quad (15)$$

The equations are similar for red channel estimation at a blue pixel location.

For red and blue channel estimation at green pixels, we employ bilinear interpolation over color differences since considered adaptive approaches do not provide any performance gain. Here, only the closest two neighbors for which the original pixel value available are used.

$$\hat{B}_{2i,2j} = G_{2i,2j} - \frac{(\hat{G}_{2i-1,2j} - B_{2i-1,2j)} + (\hat{G}_{2i+1,2j} - B_{2i+1,2j)}}{2}$$

$$\hat{B}_{2i+1,2j+1} = G_{2i+1,2j+1} - \frac{(\hat{G}_{2i+1,2j} - B_{2i+1,2j)} + (\hat{G}_{2i+1,2j+2} - B_{2i+1,2j+2)}}{2} \quad (16)$$

By the end of this step, we filled in all the missing color channel values in the input image. We utilized a simple edge strength filter both to determine the initial green channel interpolation direction and to avoid applying constant color difference rule across edge structures.

III. EXPERIMENTAL RESULTS

The proposed algorithm is tested on the Kodak image set that was used in a recent survey paper [14]. The test set consists of 12 images with 512-by-768 pixel resolution. The images are first downsampled in Bayer CFA pattern and then interpolated back to three channels using proposed algorithm. The interpolated images are compared to the original images and results are reported in terms of CPSNR error measure. Pixels within 10 pixel distance from the border are excluded

Fig. 1. image with noise

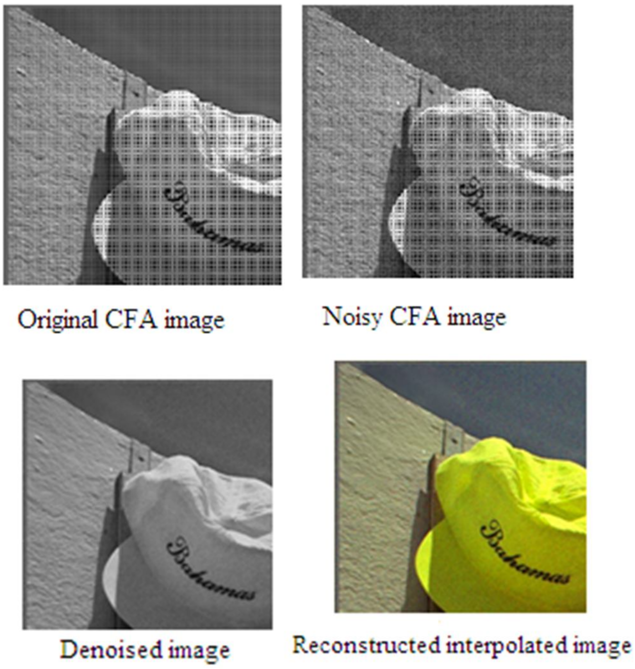

Fig. 2.CFA interpolated image

from the calculations.

The proposed method requires 376 additions, 64 multiplications, 52 absolute, 36 shifts, and 10 division operations for every 2 by 2 GRBG input pixel block. The highest performing IGD method requires between 266 and 374 operations for the same 2 by 2 block [19]. A detailed complexity comparison table can be found in [19]. A challenging image region is presented in Fig. 2 for visual quality comparison. The performance of the proposed solution under noise is compared against three highest performing methods in Fig. 3.

IV. CONCLUSION

We presented a simple edge strength filter and applied it to the CFA interpolation problem. The edge strength filter helped us identify the regions where constant color difference assumption is likely to fail which in turn lead to improved demosaicing performance. Further research efforts will focus on improving the interpolation results by exploiting spectral correlation more effectively and applying the proposed edge strength filter to other image processing problems.

ACKNOWLEDGMENT

The author would like to thank the anonymous reviewers of this paper and my supporters for their valuable and constructive review comments which helped to improve the clarity and presentation of this paper.

REFERENCES

[1] B. E. Bayer, “Color Imaging Array,” *U.S. 3 971 065*, Jul. 1976.

[2] R. Kimmel, “Demosaicing: Image reconstruction from color CCDsamples,” *IEEE Trans. Image Process., vol. 8, no. 9*, pp. 1221–1228, Sep. 1999.

[3] R. Lukac and K. N. Plataniotis, “A normalized model for color-ratio based demosaicking schemes,” in *Int. Conf. on Image Process.*, 2004, *vol. 3*, pp. 1657–1660.

[4] J. F. Hamilton and J. E. Adams, “Adaptive color plan interpolation in single sensor color electronic camera,” U.S. Patent 5 629 734, Mar. 13, 1997.

[5] C. A. Laroche and M. A. Prescott, “Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients,” U.S. Patent 5 373 322, Dec. 13, 1994.

[6] J. W. Glotzbach, R. W. Schafer, and K. Illgner, “A method of color filter array interpolation with alias cancellation properties,” in *Proc. IEEE Int. Conf. Image Process.*, 2001, *vol. 1*, pp. 141–144.

[7] 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.

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

[9] R. H. Hibbard, “Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients,” U.S. Patent 5 382 976, Jan. 17, 1995.

[10] J. E. Adams and J. F. Hamilton, Jr., “Adaptive color plan interpolation in single sensor color electronic camera,” U.S. Patent 5 506 619, Apr. 9, 1996.

[11] K.-H. Chung and Y.-H. Chan, “Color demosaicing using variance of color differences,” *IEEE Trans. Image Process., vol. 15, no. 10*, pp. 2944 2955, Oct. 2006.

[12] L. Zhang and X. Wu, “Color demosaicking via directional linear minimum mean square-error estimation,” *IEEE Trans. Image Process., vol. 14, no. 12*, pp. 2167–2178, Dec. 2005.

[13] D. Paliy, V. Katkovnik, R. Bilcu, S. Alenius, and K. Egiazarian, “Spatially adaptive color filter array interpolation for noiseless and

noisy data," *Int. J. Imag. Syst. Technol., vol. 17, no. 3*, pp. 105–122, 2007.

[14] X. Li, B. Gunturk, and L. Zhang, "Image demosaicing: A systematic survey," *Proc. SPIE–Int. Soc. Opt. Eng., vol. 6822*, p. 68221J-1-15, Jan. 2008.

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

[16] D. Menon, S. Andriani, and G. Calvagno, "Demosaicing with directional filtering and a posteriori decision," *IEEE Trans. Image Process., vol. 16, no. 1*, pp. 132–141, Jan. 2007.

[17] D. Menon and G. Calvagno, "Regularization approaches to demosaicking," *IEEE Trans. Image Process., vol. 18, no. 10*, pp. 2209–2220, Oct. 2009.

[18] I. Pekkucuksen and Y. Altunbasak, "Gradient based threshold free color filter array interpolation," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 137–140.

[19] K. H. Chung and Y. H. Chan, "Low-complexity color demosaicing algorithm based on integrated gradients," *J. Electron. Imag., vol. 19, no. 2*, p. 021104-1-15, Jun. 2010.