

Improvement of Noise free Color Video Demosaicking Using Denoising and Iterative Enhanced Demosaicking Algorithm

D.S.Binu Shoba , B.SankaraLakshmi, A.Jayachandran

*Department of CSE,
PSN College of Engineering and Technology,
Tirunelveli, India*

Abstract—Color demosaicking is critical to the image quality of digital still and video cameras that use a single-sensor array. Limited by the mosaic sampling pattern of the color filter array (CFA), color artifacts may occur in a demosaicked image in areas of high-frequency and/or sharp color transition structures. However, a color digital video camera captures a sequence of mosaic images and the temporal dimension of the color signals provides a rich source of information about the scene via camera and object motions. The quality of demosaicked images is degraded due to the sensor noise introduced during the image acquisition process. According to the proposed Demosaicking algorithm We first perform patch based denoising on the mosaic CFA video. For each CFA patch to be denoised, similar patches are selected within a local spatial-temporal neighborhood. The principal component analysis is performed on the selected patches to remove noise. we can extract edge information of each pixel in terms of the direction of variation and the gradient from the mosaic image directly and accurately, and the extracted more accurate edge information will be utilized to assist the design of our proposed new demosaicking algorithm.

Key words Color filter array(CFA),Demosaicking,Principle component analysis(PCA),Denoising,Interpolation

I. INTRODUCTION

MOST digital cameras capture a color image with a single sensor array that sub-samples color bands in a particular mosaic pattern, such as the Bayer color filter array (CFA) [3] shown in Fig. 1. At each pixel, only one of the three primary colors (red, green, and blue) is sampled. The full color image is reconstructed by interpolating the missing color samples. This process is called color demosaicking, which is critical to the quality of reconstructed color images. The problem of color demosaicking has been extensively studied in spatial and frequency domains for still digital cameras [1], [2], [4], [6], [7], [9], [11], [13]–[17], [22], [23], [26]. Early demosaicking techniques mostly work in the spatial domain, such as nearest neighbor replication, bilinear interpolation, and cubic B-spline interpolation [13]. They are easy to implement but susceptible to many artifacts such as blocking,

blurring and zipper effect at edges. The problem can be mitigated by an appropriate use of strong spectral correlation that exists in most natural color images.

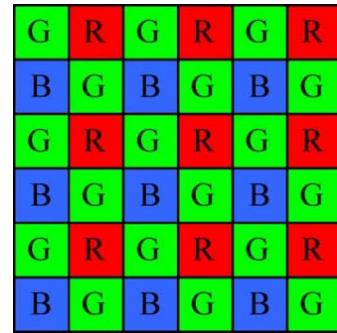


Fig. 1. Bayer pattern.

Indeed, most modern demosaicking methods exploit the correlation between red, blue, and green channels [2], [4], [6], [7], [9], [11], [13]–[17], [22], [23], [25]. To distinguish the existing methodology from the one of temporal demosaicking, we classify all the above color demosaicking techniques, either purely spatial or spatio-spectral methods, into the class of intra-frame demosaicking. Even the sophisticated spatio-spectral color demosaicking techniques, such as those recently published, can still fail to faithfully reproduce the color components in the presence of high-frequency and/or sharp color transition structures. In order to surpass the current state-of-the-art in terms of image quality, additional information and constraints of the original color signals have to be brought into the demosaicking process. For digital video cameras, the temporal dimension of a sequence of mosaic frames captures more and new data on the scene, which would be otherwise absent if only a single frame is sampled by the CFA. A natural enquiry of practical significance is how to best exploit the correlation between adjacent frames and improve the accuracy and robustness of intra-frame color demosaicking.



Fig. 2. Two stages of the proposed video denoising and demosaicking scheme for noisy CFA video

This paper presents a spatial-temporal denoising and demosaicking algorithm for noisy CFA videos. The proposed algorithm has three stages, as illustrated in Fig. 1. The first stage is to denoise the CFA video. We extend the PCA-based CFA image denoising method in [25] to a spatial-temporal one and apply it to CFA videos. Since the CFA video frames are red, green and blue color interlaced mosaic images, we consider each CFA patch, which contains pixels from all the three channels, as the basic unit in the denoising. By performing patch matching in the current frame and adjacent frames, the similar patches to the given one can be located. By viewing the elements in the CFA patch as variables and viewing the pixels in the matched patches as samples of them, the PCA transformation matrix can be computed and the denoising is then performed in the PCA domain.

In the Second stage we have proposed Demosaicking algorithm we can extract edge information of each pixel in terms of the direction of variation and the gradient from the mosaic image directly and accurately, and the extracted more accurate edge information will be utilized to assist the design of our proposed new demosaicking algorithm.

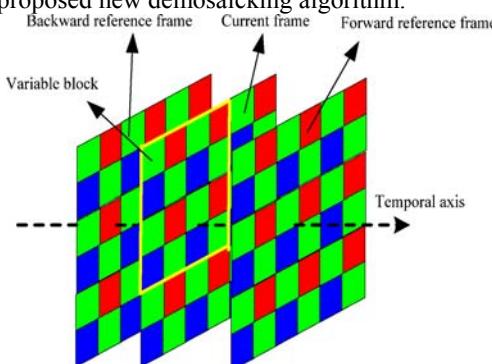


Fig. 3. CFA block to be denoised in a frame.

The proposed algorithm is also capable of reducing the interpolation errors which are visually objectionable because they tend to correlate with object boundaries and edges. Proposed algorithm find out the missing color components in the mosaic images captured by the CFA and improve the visual quality of the resulting output.

II. PCA-BASED DENOISING OF CFA MOSAIC IMAGES

Many CDM algorithms [2]–[11], [14]–[17] proposed in the past are based on the unrealistic assumption of noise-free CFA data. The presence of noise in CFA data not only

deteriorates the visual quality of captured images, but also often causes serious demosaicking artifacts which can be extremely difficult to remove using a subsequent denoising process. Note that many advanced denoising algorithms [19]–[26], which are designed for monochromatic (or full color) images, are not directly applicable to CFA images due to the underlying mosaic structure of CFAs. To overcome the problem, we will use principle component analysis (PCA)-based denoising scheme which directly operates on the CFA domain of captured images. The important property of PCA is that it is optimal by using a subset of its principal components to represent the original signal. For example, one could use the first k most important eigenvectors to form the transformation matrix as

$$\mathbf{P}^T = [\phi_1 \ \phi_2 \ \dots \ \phi_k], k < m$$

The optimal dimensionality reduction property of PCA can be used for noise removal. Generally speaking, the energy of a signal will concentrate on a small subset of the PCA transformed dataset, while the energy of noise will evenly spread over the whole dataset. Therefore, by preserving only the most important subset of the transformed dataset and then conducting the inverse PCA transform, the noise could be significantly reduced while the signal being well recovered.

$\Omega(v) = E[V V^T] = \text{diag}\{\sigma_g^2, \sigma_r^2, \sigma_b^2, \sigma_g^2\}$
The optimal dimension reduction property of PCA can be used to reduce noise. By computing the covariance matrix $\bar{\mathbf{X}}$ of $\bar{\mathbf{x}}$, the optimal PCA transformation matrix $\mathbf{P}_{\bar{\mathbf{x}}}$ of $\bar{\mathbf{x}}$ can be obtained. However, the available dataset $\bar{\mathbf{x}}$ is noise corrupted so that cannot be directly computed. Fortunately, $\mathbf{P}_{\bar{\mathbf{x}}}$ can be estimated using the linear noise model $\bar{\mathbf{x}} = \bar{\mathbf{x}} + \mathbf{v}$.

Assuming that n training samples are available for each element of x the covariance matrix of x can be estimated using maximal likelihood estimation (MLE).

$$\begin{aligned} \Omega_{\bar{\mathbf{x}}} &= E[(\bar{\mathbf{x}} - E[\bar{\mathbf{x}}])(\bar{\mathbf{x}} - E[\bar{\mathbf{x}}])^T] \\ &\approx \frac{1}{n} \bar{\mathbf{X}} \bar{\mathbf{X}}^T \\ &= \frac{1}{n} (\bar{\mathbf{X}} \bar{\mathbf{X}}^T + \bar{\mathbf{X}} \mathbf{V}^T + \mathbf{V} \bar{\mathbf{X}}^T + \mathbf{V} \mathbf{V}^T). \end{aligned}$$

Since the signal X and noise V are uncorrelated, items XV^T and VX^T $\Omega_{\bar{\mathbf{x}}}$ will be nearly zero matrices which reduces the above expression of to the following:

$$\Omega_{\bar{x}} = \Omega_{\bar{x}} + \Omega_V \approx \frac{1}{n}(\bar{X}\bar{X}^T + VV^T)$$

where $\Omega_{\bar{x}} \approx (1/n)\bar{X}\bar{X}^T$ and $\Omega_V \approx (1/n)VV^T$ are the covariance matrices of \bar{x} and v , respectively.

Estimating $\Omega_{\bar{x}}$ from $\Omega_{\bar{x}}$ requires Ω_V is known. With the noise vector

$$\mathbf{v} = [v_{g_1} \ v_{r_2} \ v_{b_3} \ v_{g_4}]^T$$

$$\Omega_V = E[\mathbf{v}\mathbf{v}^T] = \text{diag}\{\sigma_g^2, \sigma_r^2, \sigma_b^2, \sigma_g^2\}$$

where σ_g , σ_r , and σ_b are the standard deviations of the channel

III. 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

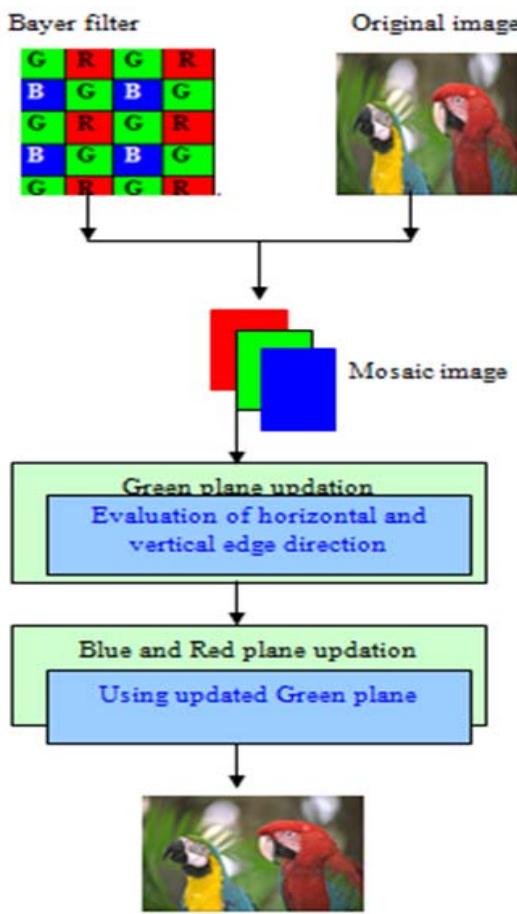


Fig.2 Demosaicing Algorithm

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 ,That is 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 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

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

Calculate The Four adjacent color difference calculated as follows

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

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 K_x , as defined In equation (2) ,The estimating $K_{x(i,j)}$ calculated as follows

$$\begin{aligned} K1 &= (K_{R,i-1,j}) / (1+\alpha_{i-1,j}) \\ K2 &= (K_{R,i+1,j}) / (1+\alpha_{i+1,j}) \\ K3 &= (K_{R,i,j-1}) / (1+\alpha_{i,j-1}) \\ K4 &= (K_{R,i,j+1}) / (1+\alpha_{i,j+1}) \\ K_{x(i,j)} &= (K1+K2+K3+K4) / ((1+\alpha_{i-1,j}) + (1+\alpha_{i+1,j}) + (1+\alpha_{i,j-1}) + (1+\alpha_{i,j+1})) \end{aligned} \quad (3) \quad (4)$$

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

$$\hat{G}_{i,j} = R_{i,j} + K_{x(i,j)} \quad (5)$$

Similarly The Green pixel value is calculated in Blue CFA components.

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.

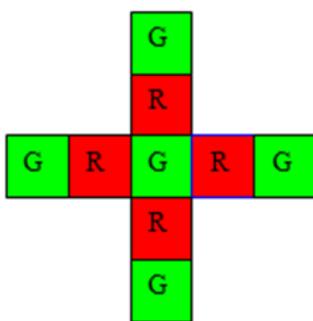


Figure 3.1 CFA pattern having their centers at Green CFA samples

As for the case in Figure 3.1 ,the missing components of the center are calculated as Follows

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

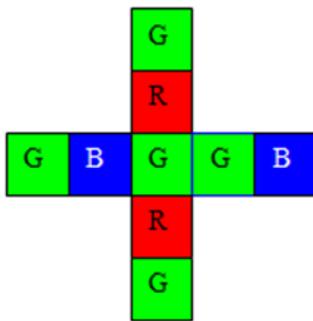


Figure 3.2 CFA pattern having their centers at Green CFA samples

As for the case in Figure 3.2 ,the missing components of the center are calculated as Follows

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

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

$$\text{col} = i \bmod 4$$

$$\text{row} = j \bmod 2$$

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

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

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

$$\hat{B}_{i,j} = \hat{B}_{i,j} + \text{nWeightage} * \hat{B}_{i,j}$$

D. 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 sampling positions are estimated using the equation given below

$$R_{i,j} = G_{i,j} + \frac{R_{i,j-1} - G_{i,j-1} + R_{i,j+1} - G_{i,j+1}}{2}$$

$$B_{i,j} = G_{i,j} + \frac{B_{i-1,j} - G_{i-1,j} + B_{i+1,j} - G_{i+1,j}}{2}$$

E. 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

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

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

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

$$\nabla g_{i,j-2} = |g_{i,j} - g_{i,j-2}|$$

$$\nabla g_{i,j+2} = |g_{i,j} - g_{i,j+2}|$$

$$b_{i,j} = g_{i,j} - (v_1(g_{i,j-1} - b_{i,j-1}) + v_2(g_{i,j+1} - b_{i,j+1})) \nabla g_{i,j-2} < \nabla g_{i,j+2}$$

$$b_{i,j} = g_{i,j} - (v_2(g_{i,j-1} - b_{i,j-1}) + v_1(g_{i,j+1} - b_{i,j+1})) \nabla g_{i,j-2} > \nabla g_{i,j+2}$$

$$b_{i,j} = g_{i,j} - (0.5(g_{i,j-1} - b_{i,j-1}) + 0.5(g_{i,j+1} - b_{i,j+1})) \nabla g_{i,j-2} = \nabla g_{i,j+2}$$

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.

IV . SIMULATION RESULTS

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

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

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.,

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

For the G channel, G0 stands for the outputs of the weighted edge interpolation, and G1 for the outputs of the color difference interpolation, respectively. As to the R and B channels, R1 and B1 are the outputs of the color-difference interpolation, but both of R0 and B0 are set to be zero. In the iteration stage, the variance of each 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 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

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

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

Where

I_0 and I_r represent, respectively, the original and reconstructed images of size $H \times W$ each.

A similar comparison study was also carried out on two real video clips. The movie sequences were originally captured on film and then digitized by a high-resolution scanner.¹ All the three color channels were known and we simulated the mosaic data by subsampling the red, green, blue channels according to the Bayer pattern. In temporal demosaicking of a current frame, we used two immediately proceeding frames and two immediately succeeding frames as reference frames. The scene in the first movie clip is a car in a park. The car is still but the camera is rotating around it. Fig. 4(a) shows the scene in a frame. The video is captured at a rate of 24-frames/second. The spatial resolution of each frame is 1948 × 1280 and the bit depth is 8 bits per color channel. In this clip, most of the smooth background objects such as the road, the lake, and trees can be reconstructed free of visible artifacts by spatial demosaicking techniques. However, on the car, where some sharp edges accompany abrupt color changes, the spatial demosaicking cannot faithfully recover the missing color components. Fig. 13(a) shows a 256 × 256 portion of the original frame in question.

Fig. 5(b)–(i) shows the demosaicked images by the eight methods. Fig. 6(a)–(i) presents the close-ups of the demosaicked images by these methods for easier visual inspection. There are highly visible color artifacts in Fig. 5(b)–(g), particularly on the grill of the car, where the true color signal frequency exceeds the sampling frequency of the Bayer CFA. The algorithm in [26] [see Fig. 4(g)] has fewer color artifacts on the grill than other intra-frame demosaicking methods, but it still generates significant zipper effects along the boundary of the red and silver colors and on the emblem of the car. The color edges that failed all intra-frame demosaicking methods have discontinuities in both luminance and chrominance. For sharp edges of highly saturated colors of weak spectral correlation, no sufficient information exists within a frame to reconstruct the color signal. This is the situation where the temporal correlation can come to the rescue. Fig. 5(h) and Fig. 6(h) are the results by the temporal demosaicking method in [24]. It offers better visual quality than intra-frame demosaicking, but some color artifacts still exist and the improvement in PSNR is quite small (referring to Table I). Fig. 5(i) and Fig. 6(i) are the demosaicked images by the proposed temporal demosaicking method. The proposed method has clear advantages over all others in terms of visual quality. Most of the color artifacts are eliminated and many sharp edge structures that are missing or distorted in intra-frame demosaicking are well reconstructed by the joint temporal-spatial demosaicking. 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.

Demosaicking Methods	Test sequence 1				Test sequence 2	
	red	green	blue	Red	green	blue
Method in [7]	31.53	33.61	27.87	33.16	38.30	36.16
Method in [4]	31.46	33.41	27.52	32.61	37.90	33.70
Method in [9]	30.60	34.08	29.24	31.76	38.42	35.72
Method in [16]	30.83	34.37	29.14	31.59	36.20	35.15
Method in [15]	30.11	33.09	28.99	31.68	36.71	35.93
Method in [26]	30.48	35.24	29.74	31.23	38.24	35.62
Method in [24]	32.24	35.15	30.10	34.15	38.11	36.06
Proposed temporal method	34.42	36.10	30.57	36.62	38.87	37.85

TABLE I
PSNR (dB) RESULTS ON THE TWO SYNTHETIC SEQUENCES BY DIFFERENT METHODS

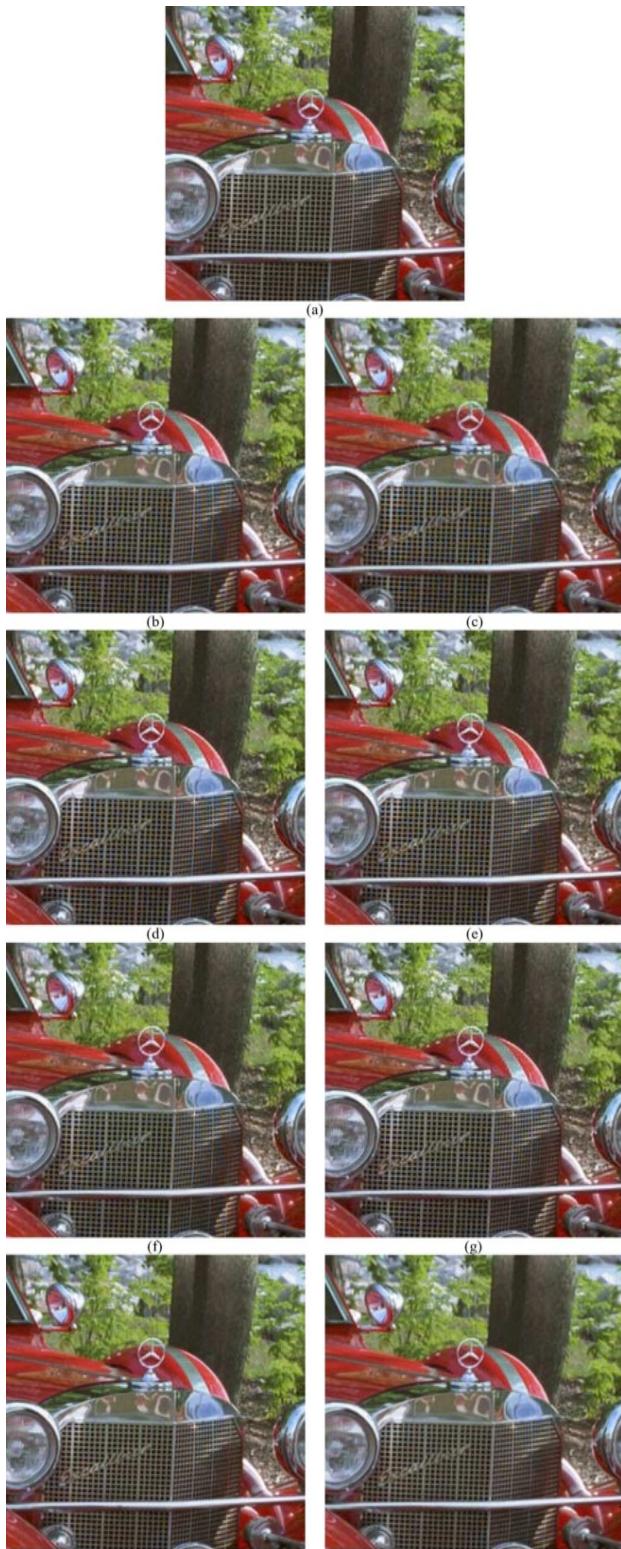


Fig. 4. (a) Original image; demosaicked images by the methods in (b) [7]; (c) [4]; (d) [9]; (e) [16]; (f) [15]; (g) [26]; (h) [24]; and (i) the proposed temporal scheme



Fig. 5 Zoom-in images of the demosaicked results. (a) Original image; demosaicked images by the methods in (b) [7]; (c) [4]; (d) [9]; (e) [16]; (f) [15]; (g) [26]; (h) [24]; and (i) the proposed temporal scheme.



Fig. 6. (a) Original image; demosaicked images by the methods in (b) [7]; (c) [4]; (d) [9]; (e) [16]; (f) [15]; (g) [26]; (h) [24]; and (i) the proposed temporal scheme.

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. 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. We can reduce the complexity by invoking the proposed new iterative demosaicing only when infra-frame demosaicking can not produce good out- puts. Only at localities of sharp edges and finely structured textures the CPU-intensive temporal color demosaicking will be activated. In smooth regions of an image, which typically constitute the major portion of a scene, the sampling frequency of the color mosaic is high enough to allow correct color demosaicing solely in the spatial domain. 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] J. E. Adams, "Intersections between color plane interpolation and other image processing functions in electronic photography," *Proc. SPIE*, vol. 2416, pp. 144–151, 1995.
- [2] "Design of practical color filter array interpolation algorithms for digital cameras," *Proc. SPIE*, vol. 3028, pp. 117–125, 1997.
- [3] B. E. Bayer, Eastman Kodak Company, Color Imaging Array, U.S.patent 3 971 065 1975.
- [4] E. Chang, S. Cheung, and D. Y. Pan, "Color filter array recovery using a threshold-based variable number of gradients," *Proc. SPIE*, vol. 3650,pp. 36–43, 1999.
- [5] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," *Proc. IEEE*, vol. 83, no. 6, pp.858–876, Jun. 1995.
- [6] 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.
- [7] J. F. Hamilton, Jr. and J. E. Adams, Adaptive Color Plane Interpolation in Single Sensor Color Electronic Camera, U.S. Patent 5 629 734 1997.
- [8] G. Jacobitti and G. Scarano, "Discrete time techniques for time delay estimation," *IEEE Trans. Signal Process.*, vol. 41, no. 2, pp. 525–533, Feb. 1993.
- [9] R. Kakarala and Z. Baharov, "Adaptive demosaicing with the principal vector method," *IEEE Trans. Consum. Electron.*, vol. 48, no. 11, pp.932–937, Nov. 2002.
- [10] S. M. Kay, *Fundamentals of Statistical Signal Processing*, 1st ed. New York: Pearson Education, Mar. 1993, vol. I, Estimation Theory. [11] R. Kimmel, "Demosaicing: Image reconstruction from CCD samples," *IEEE Trans. Image Process.*, vol. 8, no. 9, pp. 1221–1228, Sep. 1999.
- [12] P. Kuhn, *Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation*. Boston, MA: Kluwer, 1999.
- [13] P. Longère, X. Zhang, P. B. Delahunt, and D. H. Brainard, "Perceptual assessment of demosaicing algorithm performance," *Proc. IEEE*, vol.90, no. 1, pp. 123–132, Jan. 2002.
- [14] 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.
- [15] R. Lukac and K. N. Plataniotis, "Normalized color-ratio modelling for CFA interpolation," *IEEE Trans. Consum. Electron.*, vol. 50, no. 5, pp.737–745, May 2004.

- [16] 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.
- [17] R. Ramanath and W. E. Snyder, "Adaptive demosaicking," *J. Electron. Imag.*, vol. 12, no. 4, pp. 633–642, 2003.
- [18] R. R. Schultz, L. Meng, and R. L. Stevenson, "Sub pixel motion estimation for super-resolution image sequence enhancement," *J. Vis. Commun. Image Represent.*, vol. 9, pp. 38–50, Mar. 1998.
- [19] R. R. Schultz and R. L. Stevenson, "Extraction of high-resolution frames from video sequences," *IEEE Trans. Image Process.*, vol. 5, pp. 996–1011, Jun. 1996.
- [20] C. Stiller and J. Konrad, "Estimating motion in image sequence," *IEEE Signal Process. Mag.*, no. 7, pp. 70–91, Jul. 1999.
- [21] B. C. Tom and A. K. Katsaggelos, "Resolution enhancement of monochrome and color video using motion compensation," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 278–287, Feb. 2001.