AN ADAPTIVE COLOR FILTER ARRAY INTERPOLATION ALGORITHM FOR DIGITAL CAMERA

King-Hong Chung and Yuk-Hee Chan

Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong

ABSTRACT

In this paper, an adaptive color filter array (CFA) interpolation method is presented. By examining the edge levels and the variance of color difference along different edge directions, the missing green samples are first estimated. The missing red and blue samples are then estimated based on the interpolated green plane. This algorithm can effectively preserve the details as well as significantly reduce the color artifacts. As compared with some current state-of-art methods, the proposed algorithm provides outperformed results in terms of both subjective and objective image quality measures.

Index Terms—*Interpolation, charge coupled devices, cameras*

1. INTRODUCTION

Bayer color filter array (CFA), as shown in Fig. 1, is the most commonly used array in digital camera sensor (CCD or CMOS) due to its simplicity [1]. With this Bayer CFA, only one color component (R, G or B) is sampled at each pixel and, hence, the demosaicing process is required to estimate the other two missing color components for producing a full-color image [2].

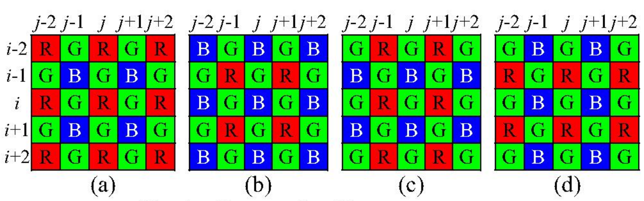

Fig. 1 Bayer color filter array pattern

In general, a demosaicing algorithm can be classified into two groups, heuristic or non-heuristic. A heuristic approach does not try to solve a mathematically defined optimization problem while a non-heuristic approach does. Methods [3-5] are examples of the non-heuristic approaches. In [3] (AP), a POCS-based algorithm is proposed where the output image is maintained within the "observation" and "detail" constraint sets while, in [4] (DUOR), an optimal-recovery-based nonlinear interpolation scheme is proposed.

As for the heuristic approach, bilinear interpolation (BI) [6] is the simplest method, in which the missing samples are interpolated on each color plane independently and details cannot be preserved well in the output image. With the use of inter-channel correlation, algorithms proposed in [7-9] attempt to maintain edge detail or limit hue transitions to provide better demosaicing performance. Algorithms [11-13] are some of the latest methods in heuristic approach. Among them, a primary-consistent soft-decision (PCSD) algorithm is proposed in [10], in which color artifacts is eliminated by ensuring the same interpolation direction for each color component of a pixel. While, in [12] (AHDDA), local homogeneity is used as an indicator to pick the direction for interpolation.

To a certain extent, it can be found that a number of heuristic algorithms, such as [10] and [12], were developed based on the framework of the adaptive color plane interpolation algorithm (ACPI) proposed in [8]. In this paper, based on the framework of ACPI, a new heuristic demosaicing algorithm is proposed. Simulation results show that the proposed algorithm is superior to the latest demosaicing algorithms in terms of both subjective and objective criteria. In particular, it can preserve the texture details in an image.

The paper is organized as follows. In Section 2, the green plane estimation in the ACPI algorithm [8] is revisited. An analysis as well as our motivation to develop the proposed algorithm is presented. Section 3 presents the details of our demosaicing algorithm, and in Section 4 some simulation results and complexity analysis are presented. Finally, a conclusion is given in Section 5.

2. OBSERVATIONS ON ADAPTIVE INTERPOLATION ALGORITHM

In ACPI [8], the green plane is first handled. For each missing green component, at position (i,j) in Fig. 1a or 1b, the algorithm performs a gradient test and then carries out an interpolation along the direction of a smaller gradient to determine the missing green component. For instance, in case of Fig. 1a, the horizontal gradient $\Delta H_{i,j}$ and the vertical gradient $\Delta V_{i,j}$ at (i,j) are determined as follows.

$$\Delta H_{i,j} = \left|G_{i,j-1} - G_{i,j+1}\right| + \left|2R_{i,j} - R_{i,j-2} - R_{i,j+2}\right| \quad (1)$$

$$\Delta V_{i,j} = \left|G_{i-1,j} - G_{i+1,j}\right| + \left|2R_{i,j} - R_{i-2,j} - R_{i+2,j}\right| \quad (2)$$

where $R_{m,n}$ and $G_{m,n}$ denote the known red and green CFA components at position (m,n). Based on these gradient values, the center missing green component $g_{i,j}$ can be interpolated by

$$g_{i,j} = \frac{(G_{i,j-1} + G_{i,j+1})}{2} + \frac{(2R_{i,j} - R_{i,j-2} - R_{i,j+2})}{4} \quad \text{if } \Delta H_{i,j} < \Delta V_{i,j} \quad (3)$$

$$g_{i,j} = \frac{(G_{i-1,j} + G_{i+1,j})}{2} + \frac{(2R_{i,j} - R_{i-2,j} - R_{i+2,j})}{4} \quad \text{if } \Delta H_{i,j} > \Delta V_{i,j} \quad (4)$$

$$g_{i,j} = \frac{(G_{i-1,j} + G_{i+1,j} + G_{i,j-1} + G_{i,j+1})}{4} + \frac{(4R_{i,j} - R_{i-2,j} - R_{i+2,j} - R_{i,j-2} - R_{i,j+2})}{8} \quad \text{if } \Delta H_{i,j} = \Delta V_{i,j} \quad (5)$$

Since the red and the blue color planes are determined based on the green plane estimation result which depends on the gradient test result, the demosaicing performance, in fact, highly relies on the success of the gradient test. To study the effect of the gradient test to the performance of the algorithm, a simple test is conducted. In the test, 24 full-color natural images, shown in Fig. 2, were sampled according to Bayer CFA pattern and then reconstructed to full-color images with ACPI [8] and the ideal ACPI respectively. The ideal ACPI is basically ACPI except that it computes all $g_{i,j}$ estimates with (3)-(5) and then picks the one closest to the real value without performing any gradient test. Note that all original images are known as reference in this test but they are not known in practice.

The peak signal-to-noise ratios (PSNRs) of their interpolated green planes are then measured. We found that the average PSNR achieved by the ideal ACPI was 43.83dB while that achieved by ACPI was only 38.18dB. Based on this result, it is observed that interpolation direction determination is critical for ACPI algorithm.

We proceeded to interpolate the red and the blue planes to produce full-color images with the same procedures mentioned in ACPI based on the two interpolated green planes. The quality of the outputs were measured in terms of the CPSNR defined in eqn.(22). As expected, the ideal ACPI achieves very high score (41.02dB) in average as compared with that achieved by ACPI (36.88dB). This shows that the approach used in ACPI to derive the other color planes with a 'good' green plane is actually very effective. As a good green plane relies on a good gradient test, the key of success is again the effectiveness of the gradient test or, to be more precise, the test for determining the interpolation direction. This finding motivates the need to find an effective and efficient method to determine the interpolation direction for improving the performance of ACPI.

Fig. 2 Set of testing images (Refers as Image 1 to Image 24, from top-to-bottom and left-to-right)

3. Proposed Algorithm

For the sake of reference, hereafter, a pixel at location (i,j) in the CFA is represented by either $(R_{i,j}, g_{i,j}, b_{i,j})$, $(r_{i,j}, G_{i,j}, b_{i,j})$ or $(r_{i,j}, g_{i,j}, B_{i,j})$, where $R_{i,j}$, $G_{i,j}$ and $B_{i,j}$ denote the known red, green and blue components and $r_{i,j}$, $g_{i,j}$ and $b_{i,j}$ denote the unknown components. The estimates of $r_{i,j}$, $g_{i,j}$ and $b_{i,j}$ are denoted by $\hat{R}_{i,j}$, $\hat{G}_{i,j}$ and $\hat{B}_{i,j}$. To get $\hat{R}_{i,j}$, $\hat{G}_{i,j}$ and $\hat{B}_{i,j}$, preliminary estimates of $r_{i,j}$, $g_{i,j}$ and $b_{i,j}$ are sometimes required during the proposed algorithm. These intermediate estimates are denoted by $\hat{r}_{i,j}$, $\hat{g}_{i,j}$ and $\hat{b}_{i,j}$.

A. Interpolating Missing Green Components

In the proposed algorithm, the missing green components are first interpolated in a raster scan manner. As mentioned before, Figs. 1a and 1b show the two possible cases and, here, the case in Fig. 1a is considered only. For the other case, the same treatment can be used by exchanging the roles of the red components and the blue components.

In Fig. 1a, the center pixel $p_{i,j}$ in the 5×5 window is represented by $(R_{i,j}, g_{i,j}, b_{i,j})$, where $g_{i,j}$ is the missing green component needed to be estimated. Instead of finding $\Delta H_{i,j}$ and $\Delta V_{i,j}$ in ACPI algorithm, edge levels, L^H and L^V, are computed in the proposed algorithm.

$$L^H = \sum_{n=-2,n\neq0}^{2} \left|A_{col}^{j+n}\right|_1^1 \quad \text{and} \quad L^V = \sum_{m=-2,m\neq0}^{2} \left|A_{row}^{i+m}\right|_1^1 \tag{6}$$

where $\left|\bullet\right|_1^1$ denotes the L_1-norm of a vector and A_{col}^{j+n} (A_{row}^{j+n}) is the difference between the $(j+n)^{th}$ and the j^{th} column (row) vectors within the window. As an example, we have $\left|A_{col}^{j+1}\right|_1^1 = \sum_{|k|\le2}\left|X_{i+k,j} - X_{i+k,j+1}\right|$, where $X_{m,n}$ is the known samples at position (m,n) in Fig. 1a. In (6), both inter-band and intra-band information are considered to detect a sharp line of width 1 pixel.

These two edge level parameters are used to estimate whether there is sharp horizontal or vertical gradient change in the window. A large value implies that there exists a sharp gradient change along a particular direction. The ratio of the two parameters is then computed to determine the dominant edge direction.

$$e = \max(L^V / L^H, L^H / L^V) \tag{7}$$

A block is defined to be a sharp edge block if $e \ge T$, where T is a predefined threshold value. In our study, experimental results show that T=2 could generally provide a promising demosaicing result. For the case of sharp edge block, the missing green component of the block center can be interpolated by using (3) or (4) after replacing $\Delta H_{i,j}$ and $\Delta V_{i,j}$ with L^H and L^V in the condition criteria of these equations respectively.

A block which is not classified to be an edge block is considered to be in a flat or texture region. In this case, as pixel color differences are more or less the same within a local region in a natural image, variance of color differences can be used as supplementary information to determine the interpolation direction for the center green components.

We extend the window size from 5×5 to 9×9 and evaluate the color differences of the pixels along the axes within the extended window. Let ${}_H\sigma_{i,j}^2$ and ${}_V\sigma_{i,j}^2$ be, respectively, the variances of the pixel color differences along the horizontal axis and the vertical axis of the window. In formulation, we have

$${}_H\sigma_{i,j}^2 = \operatorname*{var}_{|n|\le4}(\{d_{i,j+n}\}) \quad (8) \quad \text{and} \quad {}_V\sigma_{i,j}^2 = \operatorname*{var}_{|n|\le4}(\{d_{i+n,j}\}) \quad (9)$$

where $d_{p,q}$ is the color difference of a pixel at (p,q). The values of $d_{i,j+n}$ and $d_{i+n,j}$ are determined sequentially with eqns.(10)-(13) as follows.

$$d_{i,j+n} = \begin{cases} R_{i,j+n} - \hat{G}_{i,j+n} & \text{if } n = -2,-4 \\ R_{i,j+n} - \hat{g}_{i,j+n} & \text{if } n = 0,2,4 \end{cases} \tag{10}$$

$$d_{i+n,j} = \begin{cases} R_{i+n,j} - \hat{G}_{i+n,j} & \text{if } n = -2,-4 \\ R_{i+n,j} - \hat{g}_{i+n,j} & \text{if } n = 0,2,4 \end{cases} \tag{11}$$

$$d_{i,j+n} = \frac{1}{2}(d_{i,j+n-1} + d_{i,j+n+1}) \quad \text{for } n = \pm1,\pm3 \tag{12}$$

and

$$d_{i+n,j} = \frac{1}{2}(d_{i+n-1,j} + d_{i+n+1,j}) \quad \text{for } n = \pm1,\pm3 \tag{13}$$

To provide some more information about eqns.(10) and (11), we note that the missing green components are estimated in a raster scan fashion and hence the final estimates of the green components in position $\Omega_{i,j}=\{(i,j+n),(i+n,j) \mid n=-2,-4\}$ are already computed. As for the missing green components of the pixels in position $\{(i,j+n),(i+n,j) \mid n=0,2,4\}$, their preliminary estimates $\hat{g}_{i,j+n}$ and $\hat{g}_{i+n,j}$ have to be evaluated. Specifically, for n=0, 2 and 4, the $d_{i,j+n}$ for finding ${}_H\sigma^2_{i,j}$ uses the $\hat{g}_{i,j+n}$ determined with (3) unconditionally while the $d_{i+n,j}$ for finding ${}_V\sigma^2_{i,j}$ uses the $\hat{g}_{i+n,j}$ determined with (4) unconditionally.

The variance of the color differences of the diagonal pixels in the window, say ${}_B\sigma^2_{i,j}$, are determined by

$$ {}_B\sigma^2_{i,j} = \frac{1}{2}\left(\underset{|n|\le 4}{\mathrm{var}}(\{d_{i,j+n}\}) + \underset{|n|\le 4}{\mathrm{var}}(\{d_{i+n,j}\}) \right) \tag{14} $$

The same set of eqns.(10)-(13) are used to get the color difference $d_{p,q}$ required in the evaluation of ${}_B\sigma^2_{i,j}$. The only difference is that the preliminary estimates $\hat{g}_{i,j+n}$ and $\hat{g}_{i+n,j}$ involved in these equations are determined with (5) unconditionally instead of (3) and (4).

Finally, the interpolation direction for estimating the missing green component at $p_{i,j}=(R_{i,j}, g_{i,j}, b_{i,j})$ can be determined based on ${}_H\sigma^2_{i,j}$, ${}_V\sigma^2_{i,j}$ and ${}_B\sigma^2_{i,j}$. It is the direction providing the minimum variance of color difference. The missing $g_{i,j}$ can then be estimated with either (3), (4) or (5) without concerning $\Delta H_{i,j}$ and $\Delta V_{i,j}$. In other words, we have

$$ \hat{G}_{i,j} = \begin{cases} \text{unconditional result of (3) if } {}_H\sigma^2_{i,j} = \min({}_H\sigma^2_{i,j}, {}_V\sigma^2_{i,j}, {}_B\sigma^2_{i,j}) \\ \text{unconditional result of (4) if } {}_V\sigma^2_{i,j} = \min({}_H\sigma^2_{i,j}, {}_V\sigma^2_{i,j}, {}_B\sigma^2_{i,j}) \\ \text{unconditional result of (5) if } {}_B\sigma^2_{i,j} = \min({}_H\sigma^2_{i,j}, {}_V\sigma^2_{i,j}, {}_B\sigma^2_{i,j}) \end{cases} \tag{15} $$

B. Interpolating Missing Red and Blue Components at Green CFA sampling positions

After interpolating the green plane, the missing red and blue components at green CFA sampling positions are estimated by linearly interpolating the color difference planes. Figs. 1c and 1d show the two possible cases we encounter when estimating these components. For the case shown in Fig. 1c, the missing red and blue components at the center pixel are obtained by

$$ \hat{R}_{i,j} = G_{i,j} + \frac{1}{2}(R_{i,j-1} - \hat{G}_{i,j-1} + R_{i,j+1} - \hat{G}_{i,j+1}) \tag{16} $$

$$ \hat{B}_{i,j} = G_{i,j} + \frac{1}{2}(B_{i-1,j} - \hat{G}_{i-1,j} + B_{i+1,j} - \hat{G}_{i+1,j}) \tag{17} $$

As for the case shown in Fig. 1d, the center missing components are obtained by

$$ \hat{R}_{i,j} = G_{i,j} + \frac{1}{2}(R_{i-1,j} - \hat{G}_{i-1,j} + R_{i+1,j} - \hat{G}_{i+1,j}) \tag{18} $$

$$ \hat{B}_{i,j} = G_{i,j} + \frac{1}{2}(B_{i,j-1} - \hat{G}_{i,j-1} + B_{i,j+1} - \hat{G}_{i,j+1}) \tag{19} $$

C. Interpolating Missing Blue (Red) Components at Red (Blue) CFA sampling positions

Finally, the missing blue (red) components at the red (blue) sampling positions are interpolated. Figs. 1a and 1b show the two possible cases where the pixel of interest lies in the center of a 5×5 window. For the case in Fig. 1a, the missing blue sample of the center, $b_{i,j}$, is interpolated by

$$ \hat{B}_{i,j} = \hat{G}_{i,j} + \frac{1}{4}\sum_{m=\pm 1}\sum_{n=\pm 1}(B_{i+m,j+n} - \hat{G}_{i+m,j+n}) \tag{20} $$

As for the case in Fig. 1b, the missing red sample of the center, $r_{i,j}$, is interpolated by

$$ \hat{R}_{i,j} = \hat{G}_{i,j} + \frac{1}{4}\sum_{m=\pm 1}\sum_{n=\pm 1}(R_{i+m,j+n} - \hat{G}_{i+m,j+n}) \tag{21} $$

At last, the final full-color image is obtained.

D. Refinement

Refinement schemes are usually exploited to further improve the performance of the interpolation in various demosaicing algorithms [10-13]. In the proposed algorithm, we use the refinement scheme suggested in the enhanced ECI algorithm (EECI) [11] as we found that it matched the proposed algorithm to provide satisfactory demosaicing results. This refinement scheme processes the interpolated green samples $\hat{G}_{i,j}$ first to reinforce the interpolation performance and, based on the refined green plane, it performs a refinement on the interpolated red and blue samples. One can see [11] for more details on the refinement scheme.

4. Simulation Results and Complexity Analysis

Simulation was carried out to evaluate the performance of the proposed algorithm. The 24 digital color images mentioned in Section 2 were used as testing images. Nine existing demosaicing algorithms, including BI, AP [3], DUOR [4], DSA [5], ACPI [8], PCSD [10], EECI [11], AHDDA [12] and DAFD [13], were implemented for comparison. The CIELab color difference [14] between I_o and I_r were used as one of the measures to quantify the performance of the demosaicing methods, where I_o and I_r represent, respectively, the original and the reconstructed images of size $H\times W$ each. Another measure used in the evaluation is the color-peak signal-to-noise ratio (*CPSNR*) defined as

$$ CPSNR = 10\log_{10}(255^2 / CMSE) \tag{22} $$

where $CMSE = \frac{1}{3HW}\sum_{i=r,g,b}\sum_{y=1}^{H}\sum_{x=1}^{W}(I_o(x,y,i) - I_r(x,y,i))^2$ and $I(x,y,i)$ denotes the intensity value of the i^{th} color component of the $(x,y)^{th}$ pixel of image I.

Table 1 tabulates the performance achieved by different demosaicing algorithms. It shows that the proposed algorithm produces the best average performance, in terms of both quality measures, among the tested algorithms.

Fig. 3 shows part of the demosaicing results of Image 19 for comparison. One can see that the proposed algorithm, even without applying the refinement process, can preserve the texture patterns and, accordingly, produce less color-shift artifact. These results also reflect that the proposed approach for estimating the interpolation direction is robust and works well even in pattern regions as compared with the original ACPI algorithm.

Table 2 summarizes the complexity required by the proposed algorithm in terms of number of addition (ADD), multiplication (MUL), bit-shift (SHT) and comparison (CMP). Note that some intermediate computation results can be reused during demosaicing and this was taken into account when the complexity of the proposed algorithm was estimated. Its complexity can be reduced by simplifying the estimation of ${}_H\sigma^2_{i,j}$, ${}_V\sigma^2_{i,j}$ and

${}_B\sigma^2_{i,j}$. In particular, (8), (9) and (14) can be simplified by replacing the support region $|n|\leq 4$ with $n_s=\{0,\pm 2,\pm 4\}$ and using absolute distance instead of square distance. Some demosaicing performance is sacrificed due to the simplification. The simplified version provided a CPSNR of 39.89 dB and a CIELAB color difference of 1.601 in our simulations. Its complexity is also shown in Table 2.

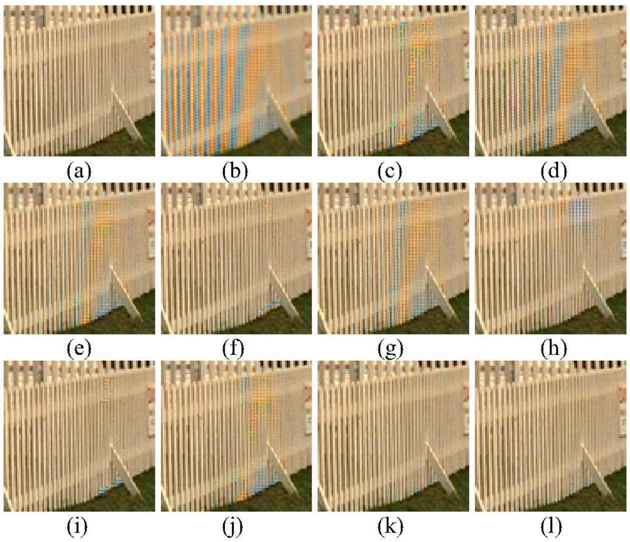

Fig. 3 Part of the demosaicing results of Image 19: (a) the original, (b) BI, (c) ACPI, (d) DAFD, (e) AP, (f) PCSD, (g) EECI, (h) DUOR, (i) AHDDA, (j) DSA, (k) the proposed algorithm without refinement and (l) the proposed algorithm with refinement

Operations per pixel	Original version				Simplified version			
	ADD	MUL	SHT	CMP	ADD	MUL	SHT	CMP
At R/B sample position								
In edge block	56	15	6	2	56	15	6	2
In non-edge block	142	54	12	4	106	19	8	4
At G sample position	38	18	2	0	38	18	2	0

Table 2 Arithmetic operations required by the proposed algorithm for estimating two missing color components at different sampling positions (including refinement).

5. CONCLUSIONS

In this paper, an adaptive demosaicing algorithm is presented. It makes use of the variances of the pixel color differences along the horizontal and the vertical axes to estimate the interpolation direction for interpolating the missing green samples. With such an arrangement, more fine texture pattern details can be preserved and a result of very little color artifact is produced in the output. Simulation results show that the proposed algorithm can produce a better demosaicing performance, both subjectively and objectively, as compared with some advanced algorithms.

6. ACKNOWLEDGMENT

This work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (PolyU 5205/04E). The authors would like to thank Dr. B. K. Gunturk, Dr. X. Li and Dr. R. Lukac for providing source code of their demosaicing algorithms [3,5,13].

REFERENCES

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

[2] B. K. Gunturk, J. Glotzbach, Y. Altunbask, R. W. Schafer, and R. M. Mersereau, “Demosaicing: color filter array interpolation”, IEEE Signal Processing Magazine, vol.22, no. 1, p.44-54, Jan. 2005.

[3] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, “Color plane interpolation using alternating projections”, IEEE Trans. Image Processing, vol.11, no. 9, p.997-1013, Sept. 2002.

[4] D. D. Muresan and T. W. Parks, “Demosaicing using optimal recovery”, IEEE Trans. Image Processing, vol.14, no. 2, p.267-278, Feb. 2005.

[5] X. Li, “Demosaicing by successive approximation”, IEEE Trans. Image Processing, vol.14, no. 3, p.370-379, Mar. 2005.

[6] T. Sakamoto, Chikako Nakanishi and Tomohiro Hase, “Software pixel interpolation for digital still camera suitable for a 32-bit MCU”, IEEE Trans. Consumer Electronics, vol. 44, no. 4, p.1342-1352, Nov. 1998.

[7] W. T. Freeman, “Median filter for reconstructing missing color samples”, U.S. Patent 4 724 395 1988.

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

[9] S. C. Pei and I. K. Tam, “Effective Color Interpolation in CCD Color Filter Arrays Using Signal Correlation”, IEEE Trans. Circuits and Systems for Video Technology, vol.13, no. 6, p.503-513, Jun. 2003.

[10] X. Wu and N. Zhang, “Primary-consistent soft-decision color demosaicing for digital cameras”, IEEE Trans. Image Processing, vol.13, no. 9, p.1263-1274, Sept. 2004.

[11] L. Chang and Y. P. Tam, “Effective use of Spatial and Spectral Correlations for Color Filter Array Demosaicing”, IEEE Trans. Consumer Electronics, vol.50, no. 1, p.355-365, Feb. 2004.

[12] K. Hirakawa and T. W. Parks, “Adaptive homogeneity-directed demosaicing algorithm”, IEEE Trans. Image Processing, vol.14, no. 3, p.360-369, Mar. 2005.

[13] R. Lukac and K. N. Plataniotis, “Data-adaptive filters for demosaicking: A framework”, IEEE Trans. Consumer Electronics, vol. 51, no. 2, p.560-570, May 2005.

[14] K. McLaren, “The development of the CIE 1976 (L*a*b*) uniform color-space and colour-difference formula”, Journal of the Society of Dyers and Colourists 92, pp. 338-341, 1976.

Methods	**Non-heuristic methods**			**Heuristic methods**								
	AP [3]	DUOR [4]	DSA [5]	BI [6]	ACPI [8]	DAFD [13]	PCSD [10]	EECI [11]	AHDDA [12]	Ours o/w Refine.	Ours /w Refine.	Ideal ACPI
Avg. CPSNR	39.15	37.70	39.12	30.06	36.88	38.28	38.83	39.61	37.98	38.48	**39.93**	41.02
Avg. Δ-CIELab	1.721	1.962	1.728	4.156	2.160	1.876	1.740	1.614	1.709	1.837	**1.593**	1.395

Table 1 The averaged CPSNR and the averaged CIELab color difference of various algorithms