DIRECTIONAL WEIGHTING-BASED DEMOSAICKING ALGORITHM FOR NOISY CFA ENVIRONMENTS[*]

Hung-Yi Lo[1], Tsung-Nan Lin[1, 2], Chih-Lung Hsu[2], and Cheng-Hsien Lee[1]

[1]Department of Electrical Engineering, [2]Graduate Institute of Communication Engineering
National Taiwan University, Taipei, Taiwan

ABSTRACT

Captured CFA data by image sensors like CCD/or CMOS are often corrupted by noises. To produce high quality images acquired by CCD/CMOS digital cameras, the problem of noise needs addressing. In this paper, we propose a novel demosaicking algorithm with the ability to handle noisy CFA data directly. By utilizing the proposed spatial filter which can characterize the similarity likelihood in local structure accurately, the noisy pixel is then filtered depending on the degree of similarity between the current pixel and a weighted average of its neighboring pixels. Therefore the edge information can be preserved without the blurring artifacts while the capacity of noise reduction can be adjusted to the maximum degree in the smooth region. Our algorithm is the first one that can accomplish the demosaicking processing and noise removal simultaneously, which contributes to the reduction of hardware cost since one module can achieve two functions efficiently at the same time.

1. INTRODUCTION

Most digital still cameras capture images by using color filter array (CFA), organized in a mosaic pattern. The most commonly used CFA is Bayer pattern [1] (Fig. 1). Observe that green pixels are sampled twice than red or blue ones. To obtain full-resolution color images, missing color pixels must be estimated from neighboring captured data. This process is known as demosaicking.

Various demosaicking techniques have been proposed [2-7,14]. All of these techniques observed the drawbacks of using the traditional bilinear interpolation, which introduces visible color bleeding artifacts, referred to as the zipper effect, around sharp or fine edges. Zipper effect is caused when the used demosaicking scheme cannot adaptively estimate the missing pixels according to the local image structure. Schemes [12] are proposed to exploit spatial correlation of images and to interpolate missing colors by a weighted-sum approach. The weights should be adaptively changed according to the image local structure as the edge indicator. However existing methods are capable of detecting either vertical or horizontal edge but fail to detect other sophisticated edges efficiently. In addition, existing methods' discussion is limited to the assumption of

noise-free environment, which is evidently not the case in real situations. To produce high quality images acquired by CCD/CMOS digital cameras, the problem of noise needs addressing. Noise is caused mainly by mean dark current and shot/read noise produced during the exposure process. Basically the statistic characteristic of such noise can be modeled as zero mean additive white Gaussian noise (AWGN), as pointed by Khaled et al [8].
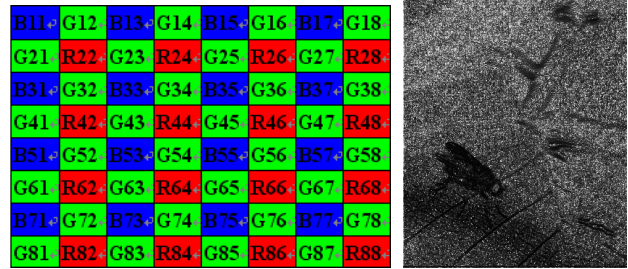

Figure 1: Bayer CFA pattern. Figure 2 Adaptiveness of the noise reduction filter.

Kalevo [10] tried to use multistage median filter and median rational hybrid filters to solve this problem. However, their approach did not take the demosaicking algorithm into account and cannot generate high-quality color images.

In this paper, our focus is geared toward designing a demosaicking algorithm with the ability to handle noisy CFA data directly. We have previously reported [9] an adaptive spatial filter that can exploit local image structures more efficiently compared to other existing methods [2-5]. We extend previous filter design to estimate the missing color pixels by taking more captured pixels into account. The accuracy of the estimation can be improved accordingly because more information is utilized. Additionally, the weighting function is modified based on the noise statistical characteristics [12]. Therefore, to our best knowledge, our algorithm is the first one that can accomplish the demosaicking processing and noise removal simultaneously, which contributes to the reduction of hardware cost since one module can achieve two functions efficiently at the same time. Simulation results demonstrate the effectiveness of the proposed algorithm.

2. AN ADAPTIVE FILTER OF DETECTING ALL

ORIENTATION EDGES

2.1. Non-cross Directional Weighting

We have previously reported [9] a spatial filter that is capable of accurately detecting complicated local image structures. Unlike traditional approaches [3, 11], which can only detect either horizontal or vertical edges, our filter can detect edges with all orientations efficiently. Our approach is based on a non-cross weighting mechanism. The weighting function should accurately estimate the similarity between the missing color pixel and its neighbors. [9] proposes the function should be inversely proportional to the Euclidean distance between the interpolated pixel and its neighboring pixels in this direction.
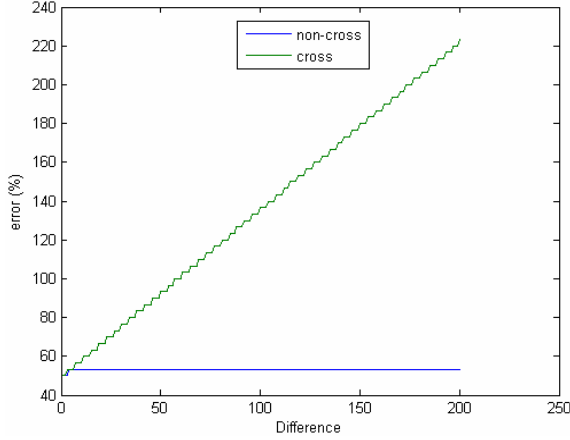

Figure 3. Error comparison between cross and non-cross method, x-axis is the difference between two sides of a 45 degree edge.

Consider missing pixel G44 is to be interpolated. Our edge detector in north direction is specified as $1/\sqrt{\text{var}(R44, R24) + \text{var}(G34, G14)}$; where var(a,b) measures the variance between a and b in statistical way. That is, $(a - \mu)^2 + (b - \mu)^2$, μ is the arithmetic mean between a and b.

Fig. 3 shows the error analysis for a CFA image with the simple geometry of a diagonal edge. Traditional filters [3, 11] work well on vertical and horizontal edges; however, the estimated error is linearly proportional to the image gradient as shown in Fig. 3. On the other hand, the proposed filter [9] is capable of detecting local structures with all orientations efficiently.

2.2. The Proposed Algorithm

It has been reported [10] previously that red and blue channels are highly correlated to green channels in real word images. That is, the contrast of the color difference plane is quite flat over the local image region. Based on this observation, it is reasonable to estimate the missing pixels by the weighted-sum approach in the color difference domain instead of the original pixel value domain.

  Step 1: Green plane interpolation: In Bayer pattern,

green pixels (the luminance channels) are sampled at higher rate than red or blue pixels (the chrominance channels). This is because the human visual system is more sensitive to the luminance information than to the chrominance. To better estimate the missing G values, our method differs from previous approaches by incorporating more information from a 5x5 kernel since neighboring pixels possess strong correlation within a local image. To estimate G(m, n) on an existing red pixel R(m,n), let's denote i the ith of the eight directions around the center pixel R(m,n) (north, south, east, west, northwest, northeast, southwest, and southeast for i = 1 to 8). Then the weighing function associated with each direction i can be calculated as:

$$W_i = \begin{cases} \dfrac{1}{\sqrt{1 + \text{var}(R(m,n), R_i) + \text{var}(G_{i1}, G_{i2})}}, i = 1 \sim 4 ...(1) \\ \dfrac{1}{\sqrt{1 + \text{var}(R(m,n), R_i) + \text{var}(B_{i1}, B_{i2})}}, i = 5 \sim 8 ...(2) \end{cases}$$

R_i denotes the pixel $R(m+v_i, n+h_i)$ where the relative position (v_i, h_i) is listed in Table 1. Similar notation for G and B pixels can also be found in Table1.

After the weights in all of the eight directions are calculated, the planar expected value within the *5x5* kernel can then be calculated as

$$\overline{G} = \frac{(\sum_{i=1}^{4} W_i * G_{i1}) + (\sum_{i=5}^{8} W_i * (\frac{G_{i3} + G_{i4}}{2}))}{\sum_{i=1}^{8} W_i} \quad (3)$$

$$\overline{R} = \frac{\sum_{i=1}^{8} W_i * R_i}{\sum_{i=1}^{8} W_i} \qquad (\overline{B} = \frac{\sum_{i=5}^{8} W_i * B_{i1}}{\sum_{i=5}^{8} W_i}) \quad (4)$$

The coefficient W indicates the degree of similarity for each captured data within the 5x5 window. The expected value can be regarded as the arithmetic output of a geometric adaptive filter which captures the local characteristics.

$$\hat{G}44 = R44 + (\overline{G} - \overline{R}) \quad (5)$$

(v_i, h_i)	R_i	G_{i1}	G_{i2}	G_{i3}	G_{i4}	B_{i1}	B_{i2}
1	(-2, 0)	(-1, 0)	(-3, 0)	---	---	---	---
2	(2, 0)	(1, 0)	(3, 0)	---	---	---	---
3	(0, 2)	(0, 1)	(0, 3)	---	---	---	---
4	(0,-2)	(0,-1)	(0,-3)	---	---	---	---
5	(-2,-2)	---	---	(-1,-2)	(-2,-1)	(-1,-1)	(-3,-3)
6	(-2, 2)	---	---	(-1, 2)	(-2, 1)	(-1, 1)	(-3, 3)
7	(2,-2)	---	---	(1,-2)	(2,-1)	(1,-1)	(3,-3)
8	(2, 2)	---	---	(1, 2)	(2, 1)	(1, 1)	(3, 3)

Table 1. The positions of nearby pixels used for directional weighting calculation in Eq. 3~5.

  Step 2: Interpolate R values on existing B pixels, and B values on existing R pixels:

Use equation (2), and then use the following equation to find B(m,n) on existing R(m,n) :

$$B(m,n) = G(m,n) - \frac{\sum_{i=5}^{8} W_i \cdot (G_{i5} - B_{i1})}{\sum_{i=5}^{8} W_i} \qquad (6)$$

where G_{i5} have the same values as B_{i1} for all i. Existing blue pixels follow similar way.

Step 3: Interpolate R and B values on existing green pixels: Find R(r,s) and B(r,s) on G(r,s).

$$R(r,s) = G(r,s) - \frac{\sum_{i=1}^{4} W_i \cdot (G_{i1} - R_{i2})}{4} \qquad (7)$$

The values of v and h for G_{i1} are the same as listed in Table 1. R_{i2} has the same value as G_{i1}. Blue value follows the same way.

3. NOISY BAYER IMAGE

3.1. Noise Removal in CFA

Captured CFA data by image sensors like CCD/or CMOS are often corrupted by noises. Noises are mainly brought about by the dark current, read noise, and the variation of pixel output voltage associated with the variation in the diode capacitance [8]. Such a disturbance is commonly referred as granular noise. Basically the statistical characteristic of such noise can be modeled as zero mean additive white Gaussian noise [12] as follows

$$\eta(z) = \frac{1}{\sigma\sqrt{2\pi}} \exp[\frac{-(z)^2}{2\sigma^2}] \qquad (8)$$

, where z is the image level and σ is the standard deviation.

The simplest noise-reduction filter is the arithmetic mean filter. However, the smooth filter degrades the image quality by blurring the sharp edges and fine details. In this paper, we propose an adaptive noise removal filter. The weighting function introduced in Section 2.2 accurately characterizes the similarity likelihood in the local structure. The noisy pixel is then filtered depending on the degree of similarity between the current pixel and a weighted average of its neighboring pixels. Additionally, the noise-reduction process is applied directly in the captured CFA data as follows:

$$\hat{X}(i,j) = X(i,j) - W(i,j)(X(i,j) - \overline{M}) \qquad (9)$$

Where X is the captured color pixels (R,G, B) in CFA data, W is the weighting function calculated according to Eq.(1)~(2), and $\overline{M}$ is the planar expected value calculated according to Eq.(3)~(5). From Eq. 9, it is easily observed that the proposed filter is equivalent to the arithmetic smooth filter when the $W(i,j)$ is equal to one, which occurs when the pixel X and planar expected value $\overline{M}$ have high degree of similarity.

Since the similarity of the local image structure is based on the weighting function, it is important to modify the function according to the statistical characteristic of the granular noise. According to Eq. 1 and 2, the value of the weight is calculated based on the distance function between $R(m,n)$ and R_i in a noise-free environment. However, in a noisy environment, the distance may not be zero, due to the noise, even when the two pixels are the same. Therefore, the curve of the weighting function should be increased to compensate the phenomenon that the distance is caused by the noise, not the image structure. The adaptive approach of the proposed filter is clearly visible in Fig. 2 where the light gray zones represent heavily filtered areas and the dark gray zones represent lightly filtered areas. After the noisy CFA data is filtered, the 24-bit full resolution image can be obtained based on the procedures of Sec 2.2.

4. EXPERIMENTAL RESULTS

Twenty benchmark images from the Kodak photo sampler are evaluated for both noise-free and noisy environments. These images are first sampled with Bayer CFA to produce mosaic images, and then used as the input for interpolation. For noise simulation, the noise (with standard deviation from 1 to 15) is added into the CFA data. In the noise-free environments, the PSNR results and its comparison with other existing methods [5] are shown in Table 2. We compare results of3 best pre-filter mechanisms discussed in [10] for the noisy environments. Fig 4 shows the proposed mechanism outperforms the methods [10] in all *R, G, B* planes in the noisy environments. The results of real images comparison with the noise (σ=10) are demonstrated in Fig. 5 and Fig. 6.

5. CONCLUSION

In this paper, we propose a novel demosaicking algorithm with the ability to handle noisy CFA data directly. By utilizing the proposed spatial filter which can characterize the similarity likelihood in local structure, the noisy pixel is then filtered depending on the degree of similarity between the current pixel and a weighted average of its neighboring pixels. Therefore the edge information can be preserved without the blurring artifacts while the capacity of noise reduction can be adjusted to the maximum degree in the smooth region. Simulation results demonstrate the effectiveness of the proposed mechanism for both noise-free and noisy environments.

REFERENCES
[1] B. E. Bayer, "Color Imaging Array," *U.S.Patent*, no. 3,971,065, 1976.
[2] S.C. Pei and I.K. Tam, "Effective Color Interpolation in CCD Color Filter Arrays Using Signal Correlation," *IEEE Transactions on Circuits and Systems for Video Technology,* Vol.13, pp. 503-513, No.6, June 2003.
[3] Lukac, R.; Plataniotis, K.N., "Color filter arrays: design and performance analysis," IEEE Transactions on Consumer Electronics, Volume 51, Issue 4, pp.1260 - 1267, Nov. 2005.
[4] Xiaomeng Wang, Weisi Lin, and Ping Xue;, "Demosaicing with Improved Edge Direction Detection," *IEEE International Symposium on Circuits and Systems,* pp. 2048-2051, May 2005.

Image	Bilinear	ECI	NEDI	AP	YPT	Proposed	Image	Bilinear	ECI	NEDI	AP	YPT	Proposed	Image	Bilinear	ECI	NEDI	AP	YPT	Proposed	Image	Bilinear	ECI	NEDI	AP	YPT	Proposed
1	25.31	33.21	33.77	37.17	37.93	39.219	6	26.7	34.65	35.59	38.34	38.82	39.838	11	28.17	35.62	34.98	38.66	38.33	40.165	16	30.3	37.51	39.41	41.46	42.5	42.657
	29.58	35.65	35.95	40.02	41.6	42.088		31.05	36.98	37.67	41.21	41.62	42.75		32.22	38.06	37.7	40.93	41.96	43.069		34.73	40.14	41.43	44.39	45.38	45.61
	25.35	33.38	34.33	36.96	38.61	39.224		27	34.12	35.17	37.07	37.65	38.668		28.34	36.09	35.32	38.36	39.53	40.015		30.39	37.32	39.3	40.46	41.8	41.513
2	31.89	36.85	37.24	38.29	37.72	40.059	7	32.57	40.31	38.58	41.85	41.86	43.491	12	32.67	39.22	40.17	42.08	41.85	43.931	17	31.63	39.14	36.53	41.48	41.14	43.12
	36.26	41.28	41.37	40.2	43.5	43.127		36.47	42.16	41.41	43.32	45.32	46.246		36.82	42.4	42.87	44.3	46.04	46.888		34.54	40.35	39.22	43.65	43.38	45.043
	32.36	39.35	39.87	38.73	41.6	40.886		32.58	39.69	38.83	39.54	41.21	41.567		32.35	39.38	40.4	41.17	42.79	43.416		30.86	38.59	36.82	40.13	40.16	41.306
3	33.45	40.51	39.57	41.29	42.23	44.003	8	22.53	30.16	30.58	35.36	35.66	37.068	13	23.09	31.43	29.52	34.4	34.25	36.977	18	27.32	34.92	33.35	36.74	35.73	38.87
	37.17	43.11	41.67	43.23	45.86	46.533		27.4	33.09	33.61	37.99	39.76	39.912		26.49	32.64	32	36.79	36.43	38.571		30.54	36.54	36.05	39.47	39.1	41.505
	33.83	40.07	39.2	39.82	41.63	43.395		22.48	29.87	30.27	34.26	35.97	36.339		23	30.64	29.37	33.01	33	34.985		26.82	34.82	34.03	35.95	36.35	37.741
4	32.61	36.88	36.89	37.7	36.98	39.343	9	31.56	38.66	39.53	41.37	41.81	43.644	14	28.15	34.78	34.41	35.81	36.45	37.644	19	26.78	34.53	36.37	39.44	39.54	41.32
	36.53	42.16	41.58	42.15	43.82	44.3		35.72	41.37	41.68	44	45.35	46.289		32.01	37.87	37.56	37.75	41.22	40.65		31.74	37.23	38.81	42.58	42.81	44.372
	32.91	40.77	40.09	41.29	42.28	43.296		31.38	39.32	40.07	40.81	42.8	42.159		28.6	35.15	34.7	34.12	36.89	36.339		26.95	34.76	37.29	39.11	40.36	40.409
5	25.75	35.04	31.45	37.7	37.53	39.369	10	31.84	39.37	38.14	41.44	41.02	43.083	15	31.05	36.33	34.76	38.02	36.65	39.704	20	30.81	38.93	37.15	41.1	41.48	43.475
	29.32	36.84	33.7	39.42	41	41.432		35.37	42.07	40.02	44.3	45.24	46.567		35.36	41.11	39.43	40.09	42.69	42.328		34.58	40.56	39.69	43.2	44.13	45.637
	25.92	34.74	31.13	35.53	36.68	37.314		31.23	39.43	38.1	40.63	41.68	42.912		32.26	39.31	37.48	38.7	40.38	41.04		30.59	37.48	36.44	38.2	39.53	40.958

Table 2. Comparison of our algorithm with existing techniques, YPT stands for Yap-Peng Tan's method in proposed in [5]. Other abbreviations can be found in [5].
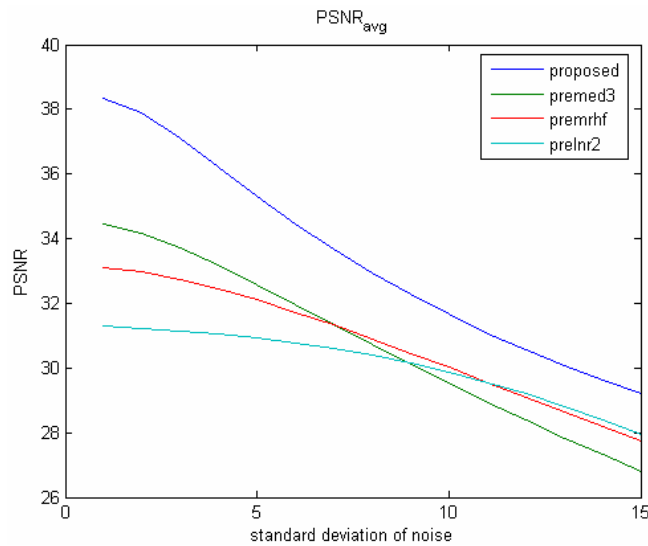

Figure 4. PSNR to standard deviation of Gaussian noise.

[5] Lanlan Chang and Yap-Peng Tan, "Adaptive color filter array demosaicing with artifact suppression," *ISCAS '04,Proceedings of the 2004 International Symposium on Circuits and Systems*, Vol. 3, pp. 937-940, May 2004.

[6] A.Lukin and D.Kubasov, "High-Quality Algorithm for Bayer Pattern Interpolation," *Programming and Computer Software*, Vol.30, No. 6, pp. 347-358, 2004.

[7] Muresan, D.D.; Parks, T.W., "Demosaicing using optimal recovery," IEEE Transactions on Image Processing, Volume 14, Issue 2, pp.267 – 278, Feb. 2005.

[8] Khaled Salama and Ahmad Al-Yamani, "Analysis of Self-Correction Active Pixel Sensors," *Proceedings of SPIE*, Vol.5668, pp. 262-269, 2005.

[9] T. Lin and C. Hsu, "Directional Weighting-Based Demosaicking Algorithm", Proceedings of 5th Pacific Rim Conference, pp. 849-857, 2004.

[10] Ossi Kalevo and Henry Rantanen, "Noise Reduction Techniques for Bayer-Matrix Images," *Proceedings of SPIE*, Vol.4669, pp. 348-359, 2002.

[11] Adams, J.E., Jr.. "Design of practical color filter array interpolation algorithms for digital cameras .2." *International Conference on Image Processing, ICIP 98,* Volume 1, 4-7, pp. 488 – 492, 1998.

[12] R. Kimmel, "Demosaicking: imge reconstruction from color CCD sample", IEEE trans. Image Processing. 7(3): 1221-1228, 1999

[13] G. Cortelazzo etal, "Statistical Characteristics of Granular Camera Noise, IEEE Trans. on Ciru. and Sys. for video tech. 4(3):536-543,1994.
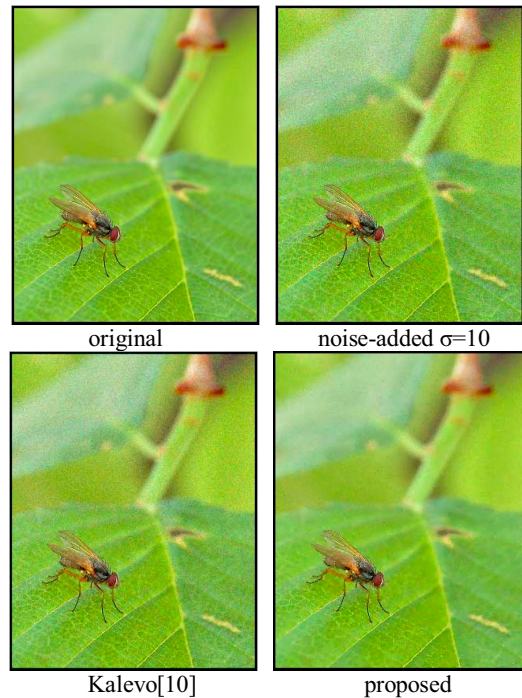
original · noise-added σ=10 · Kalevo[10] · proposed

Figure 5: An image example in noisy environments.
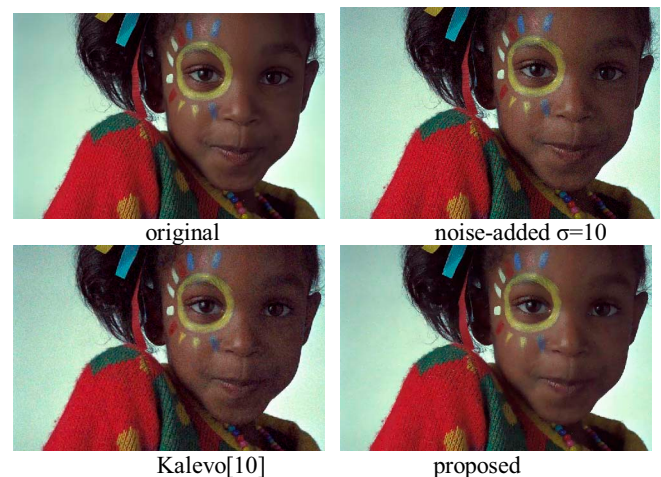

original · noise-added σ=10 · Kalevo[10] · proposed

Figure 6: Another image example in noisy environments.