Real-Time Demosaicking for Embedded Systems

Wei Hsu and Chiou-Shann Fuh

***Abstract*--We present an effective and efficient demosaicking algorithm for natural images. A new reciprocal table accurately estimates the relation between edges and gradient weights, and it achieves regular, succinct, and fast operations with excellent performance. Flexible and optional procedures adapt to different constraints on embedded systems. Our proposed method has 10.94 dB, 9.14 dB, and 9.05 dB improvements over bilinear method in red, green, and blue channels respectively. In visual results, the reconstructed images have smooth artifact. The performance is very close to recent methods, still superiorly.**

I. INTRODUCTION

Digital still cameras and camera phones are very popular today, and a sensor is a major component in captured devices. Manufacturers of sensors reduce costs and overcome techniques, and then Bayer pattern [1] is developed. A color filter of mosaic pattern covers photo detector on a sensor, and the mosaic form of filters is called Color Filter Arrays (CFA). The primary color filters are usually red (R), green (G), and blue (B) color per pixel. In each 2 by 2 pixels, green filters oppose diagonally, and the other pixels are red and blue ones. Human is sensitive to luminance information, and green is twice than R or B. Demosaicking algorithm interpolates the other two colors per pixel, and transfers mosaic images to full color images.

Imperfect demosaicking algorithms make artifact, such as, false color, zipper effect, maze, moiré, blur, and so on. A better interpolated method can reduce artifact to recover original images in vision.

A realizable algorithm has better performance and lower computational complexity in real platforms. Interpolated methods of bilinear, cubic, and smooth hue transition [2] are simple, but we are hard to accept their performance under high-quality requirements. Economically, we can apply them to low-cost sensors and systems. Recent algorithms have excellent performance, but they are too complicated. Especially in embedded systems of resource shortage, we have difficulty in implementation, and we face challenges of computational time and memory space. Our proposed algorithm really acquires the balance among performance, time, and space.

II. THE ALGORITHM

Our method is a weighted base in color-difference domain [3]. We present the single and full processes with *central interpolation* and *quarter interpolation* in Fig. 1. The single process goes through steps directly, and the full process breaks the loop as the second time. Every step refers to previous results as input. Green color has more information, and we start from the color to correct pixels step by step. For more iterative processes, it loses not only reconstructed quality but also computational performance.

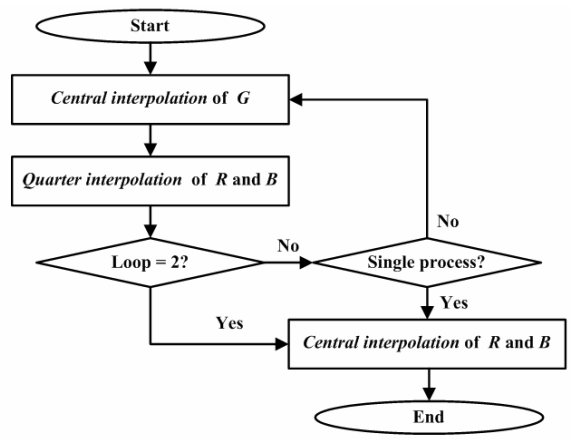
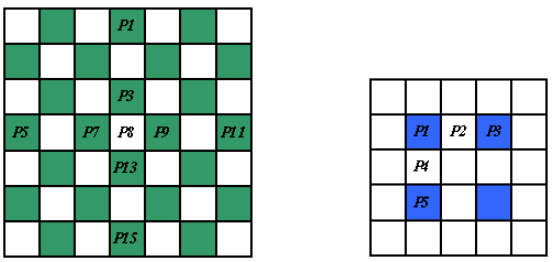


Fig. 1. Interpolated processes.

A. Central Interpolation

Central interpolation interpolates the central pixel (*P8*) of interlaced form in Fig. 2 (a).



(a) (b)

Fig. 2. The pixel form for interpolations. (a) Interlaced form for *central interpolation*. (b) Quarter form for *quarter interpolation.*

We only estimate two directional weights. Horizontal Component (*HC*) and Vertical Component (*VC*) are by

$$HC = P7 + P9 \quad (1)$$

$$VC = P3 + P13 \quad (2)$$

The perpendicular weigh-gradients (*dH* and *dV*) are arranged into the reciprocal table of 1536 words, and the equations are expressed as

$$dH = (|P7 - P11| + |P9 - P5|) >> 1 \quad (3)$$

$$dV = (|P3 - P15| + |P13 - P1|) >> 1 \quad (4)$$

The edge-weight reciprocal table represents the gradient summation on edges. By iterative experiments, we adjust reciprocal curvatures to converge the minimum interpolated errors in sample images. Reciprocal Weighted Sum (*RWS*) is

1-4244-0763-X/07/$20.00 ©2007 IEEE

by (5). Perpendicular Weight (*PW*) gives a weight between *HC* and *VC*, and it is clipped from -1 to 1 in floating-point in (6). Finally, *P8* is calculated by (7).

$$RWS = TableLookUp(dV + dH) \quad (5)$$

$$PW = (dV - dH)RWS \quad (6)$$

$$P8 = ((HC + VC) + (HC - VC)PW) >> 2 \quad (7)$$

B. Quarter Interpolation

Simply, *quarter interpolation* interpolates horizontal pixel (*P2*) and vertical pixel (*P4*) of quarter form in Fig. 2 (b).

$$P2 = (P1 + P3) >> 1 \quad (8)$$

$$P4 = (P1 + P5) >> 1 \quad (9)$$

III. EXPERIMENTAL RESULTS

We base on the 24 typical images [5] to evaluate demosaicking performance. Authors use different version images, and we cannot compare their resultant statistics directly [4]. Image sizes, compressed ratios, and compressed algorithms lead to the results very differently. For objective and general comparison, we use TIFF (Tagged Image File Format) images of 768 by 512 pixels to evaluate PSNR (Peak Signal-to-Noise Ratio).

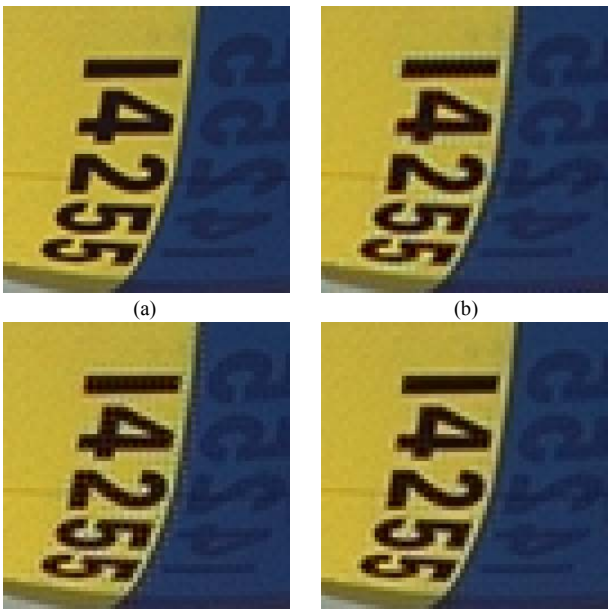
Proposed method gets the best PSNR in red channel, and adaptive frequency-domain algorithm [5] acquires the best PSNR of other channels in Table I . The differences in other channels are 0.15 dB (G) and 0.38 dB (B), but we have very low computational complexity.

TABLE I
PSNR COMPARISON

Average PSNR	Methods						
	A	B	C	D	E	F	G
R	29.28	37.26	38.43	37.03	38.85	39.33	40.22
G	33.17	38.95	41.72	41.03	42.46	40.70	42.31
B	29.22	36.04	38.49	37.08	38.65	37.68	38.27

(A) Bilinear [2]. (B) Bilinear in color-difference domain [7]. (C) Gunturk [5]. (D) Alleysson [5]. (E) Adaptive Frequency [5]. (F) Proposed method by the single process. (G) Proposed method by the full process.

PSNR and MSE (Mean-Square Error) are good statistical methods to evaluate quality of reconstructed images, but they are average results in visualization. Visual response of humans is very important equally, and we segment an image to compare with regional PSNR and artifact pattern [8]. Recent demosaicking algorithms process many textures excellently, but they make strange patterns in some specific texture. We cannot only see results by average statistics arbitrarily and directly even though PSNR has higher scores. Fig. 3 shows segmental images by original, bilinear method in color-difference, adaptive frequency-domain algorithm, and our method. Artifact, zipper effect, and false-color are obvious near edges. For artifact patterns, smooth one often can be accepted by human vision.



(a) (b) (c) (d)

Fig. 3. Image segmentation for visual and PSNR comparison (a) The original image [5]. (b) Bilinear in color-difference domain [7] with average 36.82 dB. (c) Adaptive frequency domain [5] with average 36.76 dB. (d) Our method with average 38.81 dB.

IV. CONCLUSION

By experimental results, our proposed algorithm performs not only excellent statistics but also superior vision. For flexible adjustment in resources, we lose a little performance to operate our algorithm in spatial domain. The single process saves 40 % times, and the half table saves 768 words. In different cameras, we also can calibrate the reciprocal table for better performance.

In further works, we will conform the algorithm with a suitable image pipeline for images of DSLR (Digital Single Lens), and optimize it in time and space. Noise suppression and digital zoom have the similar concept in edge-weight estimation, and we will focus on them.

REFERENCE

[1] B. E. Bayer, "Color Imaging Array," *US Patent* No.3971065, Jul. 1976.

[2] T. Chen, "A Study of Spatial Color Interpolation Algorithms for Single-Detector Digital Cameras," http://www-ise.stanford.edu/~tingchen, 2005.

[3] J. E. Adams, "Interactions between Color Plane Interpolation and Other Image Processing Functions in Electronic Photography," *Proceedings of SPIE* vol. 2416 p.144-151, 1995.

[4] E. Dubois, "Frequency-Domain Methods for Demosaicking of Bayer-Sampled Color Images," *IEEE Signal Processing Letters*, vol. 12, pp. 847-850, Dec. 2005.

[5] E. Dubois, "Frequency Domain Methods for Demosaicking of Bayer-Sampled Color Images," http://www.site.uottawa.ca/~edubois/demosaicking, Jan. 2006.

[6] T. Lin, "Demosaicking Algorithm with Noise Removal Based on Adaptive Spatial Filter," unpublished.

[7] S. C. Pei and I. K. Tam, "Effective Color Interpolation in CCD Color Filter Array Using Signal Correlation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol.13, No.6, pp.503-513, Jun. 2003.

[8] W. Hsu, "Demosaicking," http://homepage.ntu.edu.tw/~p94922001, 2006.