Edge Adaptive Color Interpolation for Ultra-Small HD-Grade CMOS Video Sensor in Camera Phones

Won-Woo Jang, Joo-Hyun Kim, Hoon-Gee Yang, Gi-Dong Lee and Bong-Soon Kang, *Member, KIMICS*

Abstract— **This paper proposes an edge adaptive color interpolation for an ultra-small HD-grade complementary metal-oxide semiconductor (CMOS) video sensor in camera phones that can process 720-p/30-fps videos. Recently, proposed methods with great image quality perceptually reconstruct the green component and then estimate the red/blue component using the reconstructed green and neighbor red and blue pixels. However, these methods require the bulky memory line buffers in order to temporally store the reconstructed green components. The edge adaptive color interpolation method uses seven or nine patterns to calculate the six edge directions. At the same time, the threshold values are adaptively adjusted by the sum of the color values of the selected pixels. This method selects the suitable one among the patterns using two flowcharts proposed in this paper, and then interpolates the missing color values. For verification, we calculated the peak-signal-to-noise-ratio (PSNR) in the test images, which were processed by the proposed algorithm, and compared the calculated PSNR of the existing methods. The proposed color interpolation is also fabricated with the 0.18-μm CMOS flash memory process.**

Index Terms— **Color interpolation, Demosaicking, Color filter array, Real-Time image signal processor.**

I. INTRODUCTION

Consumers who use high-performance cameras in mobile devices prefer video content to still-image content. In order to take video contents that is unlike still-image contents, real-time image signal processers (ISP) based on high-speed image processing are required [1]. One of the important tasks of the real-time ISP chains is Color Interpolation, which has a substantial impact on the quality of the color image [2-3].

A digital color image consists of three channels containing samples from different bands of the color spectrum, e.g., red, green, and blue. Most digital cameras, however, are equipped with a single complementary metal-oxide semiconductor (CMOS) or a charge-coupled device (CCD) image sensor and capture color images whose surface is covered with a Color Filter Array (CFA), to reduce the cost and size of the cameras [2,4]. Figure 1-(a) shows a mosaic pattern of color filters [5]. The CFA is positioned on top of the sensor to filter out the red, green, and blue components of light falling onto it. Since only one of the primary color values is recorded at the position of each pixel, the other two missing color values must be estimated from the adjacent pixel values in order to obtain a three-channel color image. This estimation of the missing color values is commonly referred to as CFA color interpolation or demosaicking. The most common CFA used today is the Bayer pattern [6]. Figure 1-(b) shows a portion of the Bayer pattern, which is widely used in digital cameras because of its excellent color signal sensitivity and good color restoration. In a Bayer pattern, the green pixels are sampled on a quincunx lattice, and the red and blue pixels are sampled on a rectangular lattice. As the Human Visual System (HVS) is more sensitive to green compared to red and blue, half of the pixels are assigned to the green pixels in the Bayer pattern [4].

There also has been growing interest in the CMOS image sensor due to the demand for a miniaturized, low-power, and cost-effective imaging system such as a digital camera, PC camera, and camcorder. The CMOS image sensor has several advantages over CCD in terms of the opportunity to integrate the sensor with other circuitry on a single chip and to reduce packaging costs. However, the image quality captured by CMOS image sensor is inferior to that by CCD [7]. Therefore, a variety of color interpolation methods have been extensively studied and proposed to improve the image quality of the CMOS image sensor.

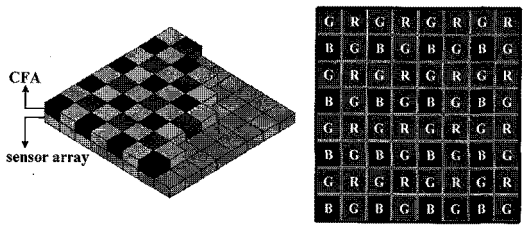


Fig. 1. (a) Bayer color filter mosaic, (b) Bayer CFA pattern.

Many demosaicking methods have been proposed. The earlier methods apply the same interpolation method to

Manuscript received January 7, 2010; revised January 18, 2010; accepted January 26, 2010.

Won-Woo Jang, Gi-Dong Lee, and Bong-Soon Kang are with the Dept. of Electronics Engineering, Dong-A University, Busan 604-714, South Korea (Tel: +82-51-200-7703, Fax: +82-51-200-7712, Email: bongsoon@dau.ac.kr)

Joo-Hyun Kim is with the SAMSUNG Electro-Mechanics Co. Ltd., Suwon 443-743, South Korea

Hoon-Gee Yang is with Dept. of Radio Science Engineering, Kwangwoon University, Seoul 139-701, South Korea

each individual color channel. These methods include nearest-neighbor interpolation, bilinear interpolation and cubic spline interpolation, and Hibbard's method. The nearest-neighbor interpolation fills in each missing color value by reproducing that from the nearest color samples [8]. The bilinear interpolation estimates missing color values using the weighted average of the neighboring color samples [8]. The cubic spline interpolation interpolates missing color values using the cubic convolution in each color channel [9]. Hibbard's method uses horizontal and vertical gradients in the green channel and chooses the interpolated green component with smaller gradients. To estimate red/green, one can use the bilinear interpolation in each channel, respectively [10].

Lately, the proposed methods exploit the color correlations to achieve better image quality perceptually and in PSNR. These methods include Laroche *et al.*, Hamilton *et al.*, and Kimmel's method. Laroche *et al.* adopt a gradient-based interpolation. This method uses horizontal and vertical gradients in the red and blue channels to reconstruct the green component with a smaller gradient. The reconstructed green components help to estimates the red/blue component. This method also reconstructs the red/blue components using reconstructed green components and neighboring red and blue pixels [11]. Hamilton's method adopts the gradients-based interpolation, which uses Laplacian second-order values and gradient values in at least two edge directions from the same column and row [12]. Kimmel's method combines a gradient-based interpolation and enhancement stage. Kimmel's gradient-based interpolation first reconstructs the green components with the help of the red and blue gradients. Then the red and blue are reconstructed using the ratio of the red/green and the blue/green, respectively. The enhancement stage involves an anisotropic inverse diffusion flow in the color space [13].

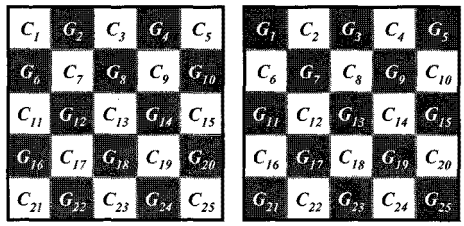
Although these methods can achieve better image quality perceptually and in the PSNR, the hardware cost for VLSI implementation is not considered in most of them. These methods can require bulky memory line buffers to store the temporally reconstructed green components [3]. Even if the methods use macro memories to avoid memory line buffers, they have the difficulty to load the camera module on a slim mobile phone due to the limitations in manufacturing technologies and also add extra cost to the camera module. To develop a cost-effective color interpolation method, the fewer the number of required memory line buffers, the better [14].

In this paper, we propose a real-time edge adaptive color interpolation method, which uses only four memory line buffers to implement the 5 × 5 array. We also designed the real-time image processing system by using Verilog-HDL for the digital video camera and camera phone that can process 720p (1280 pixels × 720 lines × 30 fps) HD videos.

The remainder of the paper is organized as follows. The proposed color interpolation method based on real-time edge adaptive color interpolation is described in Section II. The simulation result of the proposed method is presented in Section III. The hardware architecture of the proposed system is presented in Section IV, and the experimental results are shown in Section V. Conclusions are made in Section VI.

II. PROPOSED METHOD

Figure 2 illustrates a 5 × 5 array of the Bayer pattern. Figure 2-(a) shows that the central pixel is red (*R*) or blue (*B*), and Fig. 2-(b) indicates that the central pixel is green. G_i denotes the *G* pixel, and C_i designates the available *R* or *G* pixel.



(a) (b)

Fig. 2. Two types of 5 × 5 array according to the center color: (a) central pixel Red or Blue and (b) central pixel Green.

A. Figures Adaptive threshold values

We need to consider the luminance taken from a single image sensor, so we can adaptively interpolate the *G* and *R/B* in the center position. The luminance of G_i and C_i of a 5 × 5 array can be calculated as follows:

$$\begin{aligned} luminance &= \frac{C_3+C_{11}+C_{13}+C_{23}}{4} + C_{13} \quad \text{[in Fig. 2(a)]} \\ &+ \frac{G_8+G_{12}+G_{14}+G_{18}}{2} + \frac{C_7+C_9+C_{17}+C_{19}}{2} \\ &\text{or} \\ luminance &= \frac{G_7+G_9+G_{17}+G_{19}}{4} + G_{13} \quad \text{[in Fig. 2(b)]} \\ &+ \frac{C_6+C_{16}+C_{10}+C_{20}}{4} + \frac{C_6+C_{18}}{2} \\ &+ \frac{C_2+C_4+C_{22}+C_{24}}{4} + \frac{C_{12}+C_{14}}{2} \end{aligned} \tag{1}$$

From the Eq. (1), the *R*, *G*, and *B* color values are involved for calculating the luminance. Then the adaptive threshold value can be calculated as:

$$Ath = \begin{cases} Th \times p & Lumin > Max\_brightness \\ Th \times q & Min\_darkness < Lumin < Max\_darkness \\ Th & else \end{cases} \tag{2}$$

where *Th* denotes the set of the ordinary threshold values and *Ath* denotes the set of the new adaptive threshold value for a later use. *Max_brightness* represents the value of the brightness boundary. *Min_darkness* and *Max_darkness* represent the values of the minimum and maximum darkness boundary, respectively. p and q denote constants [17]. The set of the adaptive threshold value consists of six thresholds, such as Ath_{flat}, Ath_{edge}, Ath_{diag}, $Ath_{detail\_diag\_max}$, $Ath_{detail\_diag\_min}$, and Ath_{tex_mode} for later use.

B. Interpolation of missing G using flowchart with seven patterns

The proposed method interpolates the missing G at B or R in the center position as shown in Fig. 2-(a). Figure 3 shows the flowchart that classifies the current 5 × 5 array into seven patterns, such as *Horizontal*, *Vertical*, *Left Diagonal*, *Right Diagonal*, *Texture 1*, *Texture 2*, and *Flat*.

To choose the desired pattern for interpolating G, we use the sum of the absolute gradients of six directions as shown in Fig. 4. We define the edge direction parameters as the sum of the absolute difference between center identical pixel and neighborhood identical color pixels. The six parameters can be calculated as follows:

$$\begin{aligned}
\Delta H &= |2C_{13}-C_{11}-C_{15}|+|G_{12}-G_{14}| \\
&\quad +|G_8-G_6|+|G_8-G_{10}|+|G_{18}-G_{16}|+|G_{18}-G_{20}| \\
\Delta V &= |2C_{13}-C_3-C_{23}|+|G_8-G_{18}| \\
&\quad +|G_{12}-G_{22}|+|G_{12}-G_2|+|G_{14}-G_{24}|+|G_{14}-G_4| \\
\Delta RD &= |2C_{13}-C_1-C_{25}|+|C_7-C_{19}| \\
&\quad +|G_6-G_{18}|+|G_2-G_{24}|+|G_2-G_{14}|+|G_8-G_{20}| \\
\Delta LD &= |2C_{13}-C_5-C_{21}|+|C_9-C_{17}| \\
&\quad +|G_4-G_{12}|+|G_8-G_{16}|+|G_{10}-G_{18}|+|G_{14}-G_{22}| \\
\Delta DRD &= |C_{13}-C_7|+|C_{13}-C_{19}| \\
\Delta DLD &= |C_{13}-C_9|+|C_{13}-C_{17}|
\end{aligned} \tag{4}$$

where ΔH, ΔV, ΔRD, ΔLD, ΔDRD, and ΔRLD represent the values of the horizontal, vertical, right diagonal, left diagonal, detailed right diagonal, and detailed left diagonal edge direction parameters, respectively [18]. The conditions 'Flat,' 'Edge,' 'Diagonal,' 'Left,' 'Right,' 'Left_th,' 'Right_th,' 'Texture,' and 'MIN' in Fig. 3, represent Eqs. (5), (6), (7), (8), (9), (10), (11), (12), and (13), respectively.

$$(\Delta V \le Ath_{flat}) \& (\Delta H \le Ath_{flat}) \& (\Delta RD \le Ath_{flat}) \& (\Delta LD \le Ath_{flat}) \tag{5}$$

$$(|\Delta H - \Delta V| > Ath_{edge}) \& (|\Delta RD - \Delta LD| > Ath_{edge}) \tag{6}$$

$$|\Delta RD - \Delta LD| > Ath_{diag} \tag{7}$$

$$\Delta RD > \Delta LD \tag{8}$$

$$\Delta RD < \Delta LD \tag{9}$$

$$(\Delta LD > Ath_{detail\_diag\_max}) \& (\Delta RD < Ath_{detail\_diag\_min}) \tag{10}$$

$$(\Delta RD > Ath_{detail\_diag\_max}) \& (\Delta LD < Ath_{detail\_diag\_min}) \tag{11}$$

$$|\Delta DRD - \Delta DLD| > Ath_{tex\_mode} \tag{12}$$

$$\mathrm{MIN}(\Delta H, \Delta V, \Delta RD, \Delta LD) \tag{13}$$

The condition 'MIN' of Eq. (13) returns the smallest edge direction parameter among the four direction parameters. Finally, the missing G is reconstructed by the chosen pattern using Eq. (14).

$$\hat{G}_{13} = \begin{cases}
\dfrac{G_8+G_{12}+G_{14}+G_{18}}{4} & \text{if Flat} \\
\dfrac{G_{12}+G_{14}}{2} & \text{if Horizonta} \\
\dfrac{G_8+G_{18}}{2} & \text{if Vertical} \\
\dfrac{G_8+G_{12}+G_{14}+G_{18}}{4}+\dfrac{C_{13}}{2}-\dfrac{C_1+C_{11}+C_{15}+C_{25}}{8} & \text{if Right_Diagonal} \\
\dfrac{G_8+G_{12}+G_{14}+G_{18}}{4}+\dfrac{C_{13}}{2}-\dfrac{C_5+C_{11}+C_{15}+C_{21}}{8} & \text{if Left_Diagonal} \\
\dfrac{G_8+G_{12}+G_{14}+G_{18}}{4} & \text{if Texture1} \\
\dfrac{G_8+G_{12}+G_{14}+G_{18}}{4}+\dfrac{C_{13}}{2}-\dfrac{C_3+C_{11}+C_{15}+C_{23}}{8} & \text{if Texture2}
\end{cases} \tag{14}$$

where $\hat{G}_{13}$ represents reconstruction G in the center position of Fig. 2-(a).

C. Interpolation of the opposite missing R or B at each B or R using nine patterns

The proposed method interpolates the opposite missing R or B at each B or R in the center position as shown in Fig. 2-(a). Figure 5 shows the flowchart that classifies the current 5 × 5 array into eleven patterns. Except for the additional several patterns and sequences in the dotted rectangular of Fig. 5, the other patterns and sequences of the flowchart are identical to those shown in Fig. 3. In the exceptional dotted rectangular, two patterns, *Horizontal* and *Vertical*, are subdivided into additional four patterns, such as *Top Horizontal*, *Bottom Horizontal*, *Right Vertical*, and *Left Vertical*.

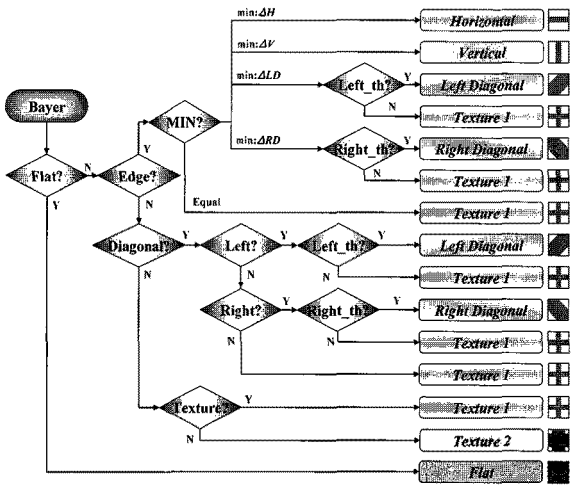


Fig. 3. The flowchart to interpolate missing G at B or R.

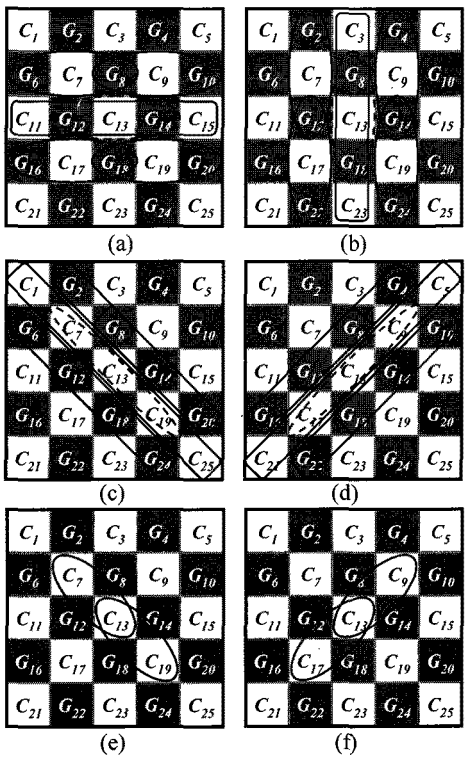


Fig. 4. Six edge direction parameters: (a) horizontal, (b) vertical, (c) right diagonal, (d) left diagonal, (e) detailed right diagonal, and (f) detailed left diagonal.

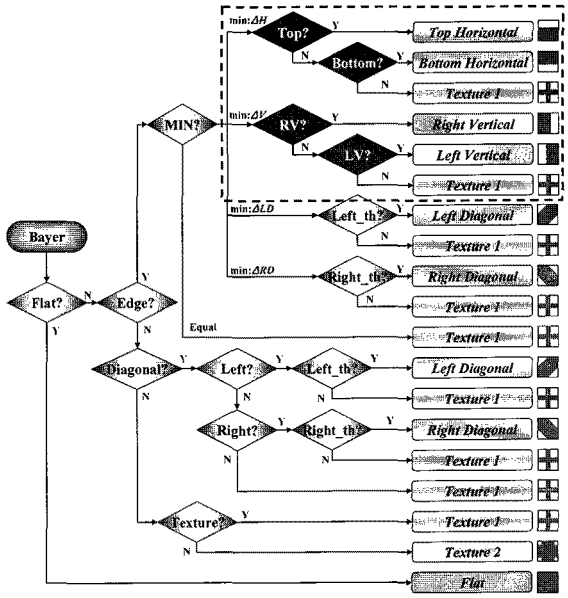


Fig. 5. The flowchart to interpolate opposite missing *R*/*B* at each *B*/*R*.

For four patterns, the additional four conditions use the absolute difference between the reconstruction *G* and neighborhood *G* pixels. Eqs. (15), (16), (17), and (18) are used for the conditions, 'Top,' 'Bottom,' 'RV,' and 'LV' of Fig. 5, respectively.

$$\left|\hat{G}_{13}-G_8\right| > \left|\hat{G}_{13}-G_{18}\right| \tag{15}$$

$$\left|\hat{G}_{13}-G_8\right| < \left|\hat{G}_{13}-G_{18}\right| \tag{16}$$

$$\left|\hat{G}_{13}-G_{12}\right| > \left|\hat{G}_{13}-G_{14}\right| \tag{17}$$

$$\left|\hat{G}_{13}-G_{12}\right| < \left|\hat{G}_{13}-G_{14}\right| \tag{18}$$

Finally, the missing *R* or *B* is reconstructed by the chosen pattern using Eq. (19).

$$\hat{C}_{13}=\begin{cases}\dfrac{C_7+C_9+C_{17}+C_{19}}{4} & \text{if Flat}\\ \dfrac{C_{17}+C_{19}}{2}-\dfrac{G_{16}+6G_{18}+G_{20}}{8} & \text{if Top Horizontal}\\ \dfrac{C_7+C_9}{2}-\dfrac{G_6+6G_8+G_{10}}{8} & \text{if Bottom Horizontal}\\ \dfrac{C_7+C_{17}}{2}-\dfrac{G_2+6G_{12}+G_{22}}{8} & \text{if Right Vertical}\\ \dfrac{C_9+C_{19}}{2}-\dfrac{G_4+6G_{14}+G_{24}}{8} & \text{if Left Vertical}\\ \dfrac{C_7+C_{13}+C_{19}}{2}-\dfrac{C_1+C_{25}}{4} & \text{if Right Diagonal}\\ \dfrac{C_9+C_{13}+C_{17}}{2}-\dfrac{C_5+C_{21}}{4} & \text{if Left Diagonal}\\ \dfrac{C_7+C_9+C_{17}+C_{19}}{4} & \text{if Texture 1}\\ \dfrac{C_7+C_9+C_{17}+C_{19}}{4}-\dfrac{G_8+G_{12}+G_{14}+G_{18}}{4} & \text{if Texture 2}\end{cases} \tag{19}$$

where $\hat{C}_{13}$ represents reconstruction *R* or *B* at each *B* or *R* in the center position of Fig. 2-(a).

D. Interpolation of missing R and B at the G

The proposed method interpolates the missing *R* and *B* at the location *G* as shown in Fig. 2-(b). Because the six edge direction parameters in Eq. (4) were calculated using identical color pixels, this proposed method uses the flowchart in Fig. 5 which is based in the modified six edge direction parameters to fix the serial numbers of all pixels and to swap G_i for C_i in Eq. (4). Also, we can get the original central pixel *G*. The condition, 'Top', 'Bottom', 'RV', and 'LV' of Fig. 5 are changed into Eqs. (20), (21), (22), and (23), respectively.

$$\left|2G_{13}-G_{17}-G_{19}\right| > \left|2G_{13}-G_7-G_9\right| \tag{20}$$

$$\left|2G_{13}-G_{17}-G_{19}\right| < \left|2G_{13}-G_7-G_9\right| \tag{21}$$

$$\left|2G_{13}-G_7-G_{17}\right| > \left|2G_{13}-G_9-G_{19}\right| \tag{22}$$

$$\left|2G_{13}-G_7-G_{17}\right| < \left|2G_{13}-G_9-G_{19}\right| \tag{23}$$

Except for the modified six edge parameters and equations matched to the four conditions, the other patterns and sequences of the flowchart are identical to those shown in Fig. 5. Because the calculations of three

methods are very similar, we can avoid using extra calculations and line buffers. Figure 6 shows the 5 × 5 array of the Bayer pattern in central pixel G. Figure 6-(a) shows that the ratio of pixels is two R and three B, and Fig. 6-(b) indicates that the ratio of pixels is three R and two B.

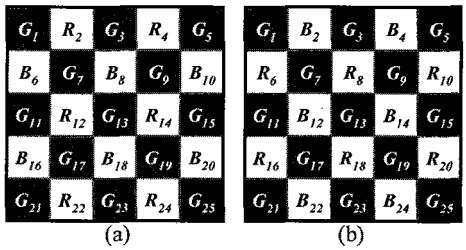


Fig. 6. Two types of 5 × 5 array in central pixel Green: (a) two R and three B and (b) three R and two B.

Finally, the missing R and B in Fig. 6-(a) are reconstructed by the chosen pattern using Eqs. (24) and (25), respectively.

$$\hat{R}_{13}=\begin{cases}\frac{R_2+R_4+R_{22}+R_{24}}{16}+\frac{3(R_{12}+R_{14})}{8} & \text{if Flat}\\ \frac{R_{12}+R_{14}}{2} & \text{if Top Horizontal}\\ \frac{R_{12}+R_{14}}{2} & \text{if Bottom\_Horizontal}\\ \frac{R_2+R_{22}+3R_{12}}{8}+G_{13}-\frac{G_7+G_{17}}{2} & \text{if Right Vertical}\\ \frac{R_{24}+R_{24}+3R_{14}}{8}+G_{13}-\frac{G_9+G_{19}}{2} & \text{if Left Vertical}\\ \frac{R_2+R_{24}+3(R_{12}+R_{14})}{8}+\frac{G_{13}}{2}-\frac{G_1+G_7+G_{19}+G_{25}}{8} & \text{if Right\_Diagonal}\\ \frac{R_4+R_{22}+3(R_{12}+R_{14})}{8}+\frac{G_{13}}{2}-\frac{G_5+G_9+G_{17}+G_{21}}{8} & \text{if Left\_Diagonal}\\ \frac{R_{12}+R_{14}}{2} & \text{if Texture1}\\ \frac{R_{12}+R_{14}}{2}+\frac{G_{13}}{2}-\frac{G_7+G_9+G_{17}+G_{19}}{8} & \text{if Texture2}\end{cases} \quad (24)$$

$$\hat{B}_{13}=\begin{cases}\frac{B_6+B_{10}+B_{16}+B_{20}}{16}+\frac{3(B_8+B_{18})}{8} & \text{if Flat}\\ \frac{B_{16}+6B_{18}+B_{20}}{8}+G_{13}-\frac{G_{17}+G_{19}}{2} & \text{if Top Horizontal}\\ \frac{B_6+6B_8+B_{10}}{8}+G_{13}-\frac{G_7+G_9}{2} & \text{if Bottom\_Horizontal}\\ \frac{B_8+B_{18}}{2} & \text{if Right Vertical}\\ \frac{B_8+B_{18}}{2} & \text{if Left Vertical}\\ \frac{B_6+B_{20}+3(B_8+B_{18})}{8}+\frac{G_{13}}{2}-\frac{G_1+G_7+G_{19}+G_{25}}{8} & \text{if Right\_Diagonal}\\ \frac{3(B_8+B_{18})}{8}+\frac{G_{13}}{2}-\frac{G_5+G_9+G_{17}+G_{21}}{8} & \text{if Left\_Diagonal}\\ \frac{B_8+B_{18}}{2} & \text{if Texture 1}\\ \frac{B_8+B_{18}}{2}+\frac{G_{13}}{2}-\frac{G_7+G_9+G_{17}+G_{19}}{8} & \text{if Texture 2}\end{cases} \quad (25)$$

where $\hat{R}_{13}$ represents reconstruction R and $\hat{B}_{13}$ represents reconstruction at the location G as shown in Fig. 6-(a). As the missing R and B in Fig. 6-(b), the serial numbers of all pixels and G_i pixels in Eqs. (24) and (25) are fixed and R_i and B_i pixels are swapped for B_i and R_i pixels, respectively.

III. SIMULATION RESULTS

Figure 7 shows test images that have been widely used for CFA demosaicking researches and experiments. To verify the performance of the proposed method, we compare our demosaicked results with those obtained by bilinear interpolation [8], Hibbard's method [10], and Laroche *et al.* [11]. The test images change for the Bayer pattern image, and then the Bayer pattern images are interpolated to reconstruct complete color image using the three methods. The interpolated images were compared to the original image to judge the interpolation quality in using the PSNR. The PSNR performances of the methods under comparison are listed in Table I. The PSNR results (in dB) of the red, green, and blue color are listed in the 1st, 2nd, and 3rd rows of each image. The best PSNR result of each row is bold. The table shows that our proposed method outperforms other methods in terms of PSNR results over most test images. In particular, the PSNR results obtained by our proposed method are, on average, 5.36 dB, 5.06 dB, and 1.15 dB better than those of the bilinear interpolation, Hibbard's, and Laroche *et al.*'s method, respectively. Also, the PSNR results of the proposed method increase by an averaged of 18 percent over those of the bilinear interpolation.

Figures 8 and 9 show parts of the test images of 15 and 19 respectively, for subjective comparison. When we compare the different methods, we can observe that fewer artifacts are generated by the proposed method than by the others.

Fig. 7. Test images.

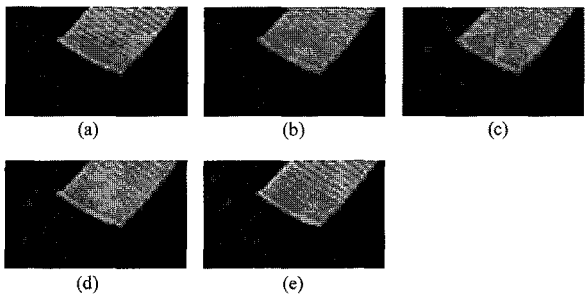


Fig. 8. Comparison different demosaicking methods for image 15: (a) Original, (b) Bilinear, (c) Hibbard, (d) Laroche, and (e) Proposed.

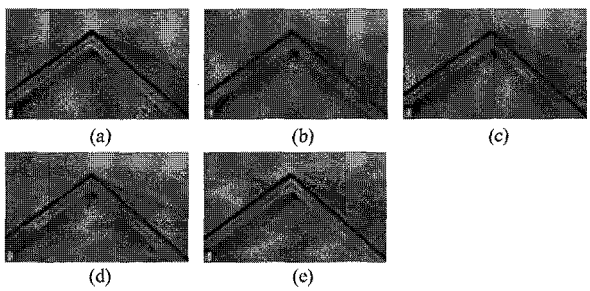


Fig. 9. Comparison different demosaicking methods for image 19: (a) Original, (b) Bilinear, (c) Hibbard, (d) Laroche, and (e) Proposed.

TABLE I
Comparisons of PSNR Performances (dB)

Image	Bilinear [8]	Hibbard [9]	Laroche [11]	Proposed	Proposed /Bilinear	Image	Bilinear [8]	Hibbard [9]	Laroche [11]	Proposed	Proposed /Bilinear
	25.15	25.15	30.83	**31.14**	1.24		23.06	23.06	26.98	**28.59**	1.24
1	28.49	29.64	31.3	**34.89**	1.22	13	25.68	25.79	27.09	**32.07**	1.25
	25.34	25.34	**31.47**	31.34	1.24		23.11	23.11	27.27	**28.31**	1.23
	32.34	32.34	**36.9**	35.60	1.10		28.32	28.32	**33.30**	33.18	1.17
2	35.73	35.97	37.27	**39.84**	1.12	14	31.14	32.06	33.33	**37.00**	1.19
	32.48	32.48	**37.26**	36.62	1.13		28.73	28.73	32.92	**33.21**	1.16
	33.45	33.45	**38.24**	37.87	1.13		31.55	31.55	35.35	**35.64**	1.13
3	36.48	37.28	38.43	**40.88**	1.12	15	34.81	35.21	36.16	**39.88**	1.15
	33.95	33.95	**37.75**	37.56	1.11		31.33	31.33	36.06	**36.76**	1.17
	32.81	32.81	36.19	**36.40**	1.11		30.28	30.28	**35.87**	35.10	1.16
4	35.91	36.12	37.35	**40.39**	1.12	16	33.29	34.3	36.56	**38.94**	1.17
	32.87	32.87	37.53	**37.59**	1.14		30.45	30.45	**36.43**	35.22	1.16
	25.6	25.6	31.25	**32.51**	1.27		31.59	31.59	35.91	**36.63**	1.16
5	28.53	29.58	30.85	**36.62**	1.28	17	34.00	34.52	35.59	**39.05**	1.15
	26.12	26.12	31.24	**32.56**	1.25		31.03	31.03	35.47	**35.87**	1.16
	26.77	26.77	**32.36**	32.24	1.20		27.28	27.28	31.36	**32.56**	1.19
6	29.76	30.71	32.83	**36.23**	1.22	18	30.06	30.04	31.29	**35.64**	1.19
	27.08	27.08	**32.65**	32.11	1.19		27.05	27.05	31.17	**32.01**	1.18
	32.57	32.57	38.42	**38.45**	1.18		26.98	26.98	**34.43**	33.23	1.23
7	35.22	36.87	38.09	**41.41**	1.18	19	30.95	33.24	34.86	**37.08**	1.20
	32.64	32.64	37.66	**38.50**	1.18		27.11	27.11	**34.81**	33.33	1.23
	22.51	22.51	**29.59**	29.38	1.31		30.82	30.82	36.08	**36.86**	1.20
8	26.52	28.51	30.19	**33.57**	1.27	20	33.64	34.89	36.01	**39.81**	1.18
	22.74	22.74	**30.02**	29.36	1.29		30.81	30.81	35.44	**35.95**	1.17
	31.49	31.49	**37.40**	37.10	1.18		27.66	27.66	32.67	**33.34**	1.21
9	34.81	36.38	37.89	**40.04**	1.15	21	30.39	31.29	32.95	**36.90**	1.21
	31.57	31.57	**37.58**	36.52	1.16		27.64	27.64	32.83	**33.17**	1.20
	31.73	31.73	**36.89**	36.80	1.16		29.95	29.95	34.31	**34.79**	1.16
10	34.43	35.81	37.27	**39.95**	1.16	22	32.76	33.29	34.54	**37.79**	1.15
	31.42	31.42	**36.83**	36.20	1.15		29.37	29.37	34.01	**34.34**	1.17
	28.2	28.2	33.38	**33.56**	1.19		34.39	34.39	**40.03**	39.42	1.15
11	31.35	32.16	33.61	**37.26**	1.19	23	37.21	38.13	39.80	**42.00**	1.13
	28.49	28.49	**33.90**	33.81	1.19		34.04	34.04	39.13	**39.40**	1.16
	32.34	32.34	**38.13**	36.10	1.12		26.39	26.39	30.12	**32.67**	1.24
12	35.81	36.84	38.79	**39.17**	1.09	24	28.79	28.9	29.94	**34.64**	1.20
	32.54	32.54	**38.22**	36.02	1.11		25.38	25.38	29.28	**31.98**	1.26
							29.30	29.30	34.42	**34.55**	1.18
						Mean	32.32	33.23	34.67	**37.96**	1.17
							29.30	29.30	34.46	**34.49**	1.18

IV. SYSTEM ARCHITECTURE

Figure 10 shows the simplified block diagram of the ISP that can be used in mobile applications. The CMOS sensor output signal is sampled at 60 MHz to meet 720P video out, and it is digitized via a 10-bit A/D converter. The digitized raw data is in the 10-bit Bayer pattern by using CFA. The pre-processing compensates for the digitized raw data and derives RGB values from the Bayer pattern signals captured from the CMOS sensor lens. The color-processing estimates missing two color values using color interpolation. The post-processing then enhances the image quality and sends 16-bit 422 YCbCr for the following display devices [19].

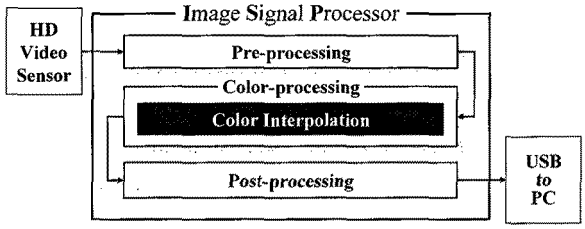


Fig. 10. Simplified block diagram of an image signal processor.

V. EXPERIMENTAL RESULTS

Figure 11 shows images captured from the prototype HD system and displayed on an HDTV Set. We used a complex image as the input scene to test the performance of the system. Figure 11-(a) and (b) show the unprocessed and processed images using the prototype HD system, which are displayed on an HDTV Set. As the green color is half of the Bayer pattern, the raw image as shown in Fig. 11-(a) appears greenish. The dotted rectangular line indicates the region for confirming the processed image and the unprocessed image. Figure 11-(c) and (e) also show the zoomed unprocessed images and Figure 11-(d) and (f) show the processed images, respectively. From the figures, we can see that the diagonal edges are clearly reconstructed by using the proposed method.

VI. CONCLUSIONS

In this paper, we proposed the edge adaptive color interpolation method and hardware implementation for cameras in mobile devices. Because the proposed system had no additional memory line buffers to store temporally green components, it is compact and efficient HD-grade video in camera phones. The proposed edge adaptive method used seven or nine patterns in the 5 × 5 array, and calculated six edge directions for considering the

horizontal, vertical, right, left diagonal, detailed right diagonal, and detailed left diagonal gradients. By doing so, the proposed method chose the desired pattern among the seven or nine patterns from the flowcharts. Compared to other methods, this method can more effectively achieve the desired effects of increased processing speed and reduced expense.

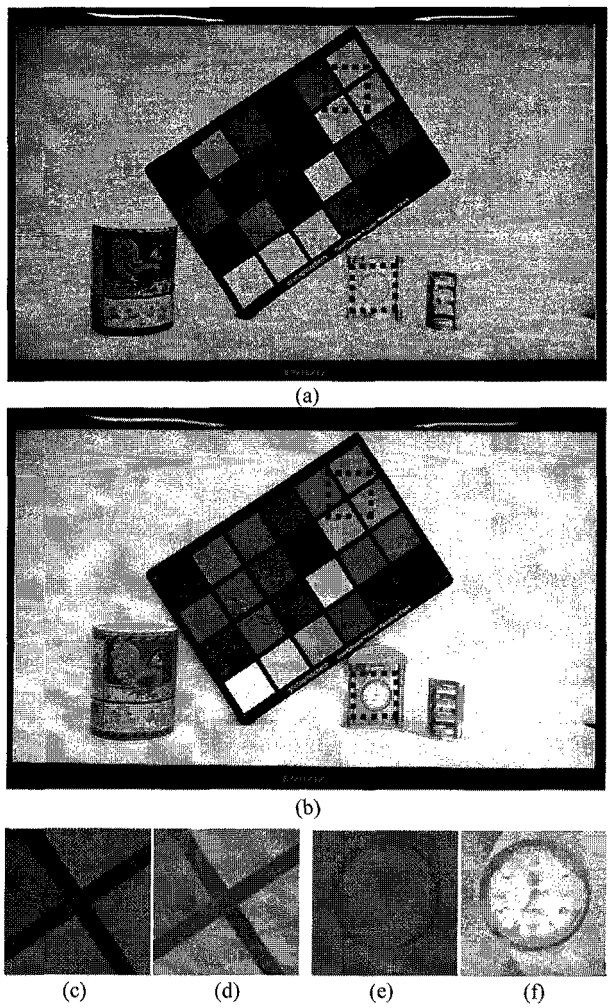


Fig. 11 Demonstration of the prototype HD system and HDTV Set: (a) unprocessed image, (b) processed image, (c) zoomed unprocessed image #1, (d) zoomed processed image #1, (e) zoomed unprocessed image #2, (f) zoomed processed image #2

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2009-0065761).

REFERENCES

[1] D. Doswald, J. Hafliger, P. Blessing, N. Felber, P. Niederer, and W. Fichtner, "A 30-frames/s megapixel real-time CMOS image processor," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 11, Nov. 2000.

[2] G. C. Holst and T. S. Lomheim, *CMOS/CCD Sensors and Camera Systems*, SPIE Press: Washington, 2007, pp. 118-121.

[3] S. C. Hsia, M. H. Chen, and P. S. Tsai, "VLSI Implementation of low-power high-quality color interpolation processor for CCD camera," *IEEE Trans. VLSI Syst.*, vol. 14, no. 4, Apr. 2006.

[4] A. C. Popescu and H. Farid, "Exposing Digital Forgeries in Color Filter Array Interpolated Images," *IEEE Trans. Signal Processing*, vol. 53, no. 10, Oct. 2005.

[5] http://en.wikipedia.org/wiki/Color_filter_array

[6] B. E. Bayer, "Color imaging array," U.S. Patent 3,971,065, Jul. 1976.

[7] Y. Jung, H. Kim, B. Hur and M. Kang, "Design of real-time image enhancement preprocessor for cmos image sensor," *IEEE Trans. Consumer Electron.*, vol. 46, no. 1, Feb. 2000.

[8] M. Unse, "Spline: A perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, vol. 16, no. 6, pp. 22-38, Nov. 1999.

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

[10] 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. 1994.

[11] J. E. Adams Jr., "Design of practical color filter array interpolation algorithms for digital camera," *Proc. SPIE*, vol. 3028, pp. 117-125, Feb. 1997.

[12] J. F. Hamilton Jr. and J. E Adams, "Adaptive color plane interpolation in single sensor color electronic camera," U. S. Patent 5,629,734, May 1997

[13] R. Kimmel, "Demosaicing: image reconstruction from CCD samples," *IEEE Trans. Image Processing*, vol. 8, pp. 1221-1228, 1999.

[14] T. H. Chen and S. Y. Chien, "Cost Effective Color Filter Array Demosaicking with Chrominance Variance Weighted Interpolation," *IEEE Int. Sym. Circuits and Syst.*, pp. 1277-1280, May 2007.

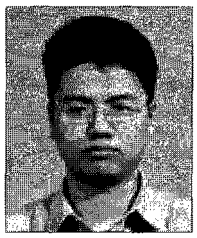
[15] S. Yun, J. Kim, and S. Kim, "Color interpolation by expanding a gradient method," *IEEE Trans. Consumer Electron.*, vol. 54, no. 4, Nov. 2008.

[16] L. Chang and Y. P. Tan, "Hybrid color filter array demosaicking for effective artifact suppression," *SPIE Journal of Electronic Imaging*, vol. 15, Mar. 2006.

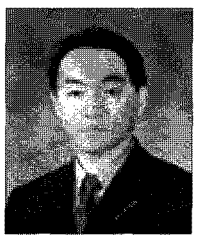
[17] H. Kim, J. Kim, W. Choi, and B. Kang, "Real-Time Edge Adaptive Color Interpolation for an Ultra Small HD-Grade Video Sensor in Mobile Devices," *IEEE Int. Conf. Signal Processing and Communication Syst.*, Dec. 2008.

[18] H. Kim, W. Jang, J. Kim, and B. Kang, "Design of improve Color Interpolation Filter for High-speed Operation and Lower Hardware-complexity," *IEEK SoC Design Tech. Society*, pp. 243-246, May 2008.

[19] W. Jang, B. Kwak, J. Kim, and B. Kang, "An Image Signal Processor for Ultra Small HD Video Sensor with 3A in Camera Phones," *IEEE Int. Conf. Consumer Electron.*, pp. 3.3-3, Jan. 2009.



Won-Woo Jang received the B.S and M.S. degrees in electronics engineering from Dong-A University, Busan, Korea, in 2005 and 2007, respectively. He is currently working toward his Ph.D. degree at the university. His research interests include digital camera processing systems, VLSI architecture design and image processing.



Joo-Hyun Kim received the B.S., M.S. and Ph.D. degrees in electronics engineering from Dong-A University, Busan, Korea, in 2002, 2004 and 2007, respectively. He is now with Samsung Electro-Mechanics Co. Ltd., Suwon, Korea. His research interests include digital camera processing systems, VLSI architecture design and image processing.

Hoon-Gee Yang received the B.S. degree in electronics engineering from Yonsei University, Seoul, Korea, in 1985 and the MS and PhD degrees in electrical and computer engineering from the State University of New York at Buffalo, Amherst, NY, in 1987 and 1992, respectively. From 1993, he has been in the Department of Radio Science and Engineering, Kwangwoon University, Seoul, Korea. His main research interests are the receiver design of the ultrawide-bandwidth communications, array signal processing and the RFID reader/tag design.

Gi-Dong Lee received the B.S., M.S. and Ph.D. degrees in electronics engineering from Pusan National University, Busan, Korea, in 1989, 1991 and 2000, respectively. He had worked at LCD R&D center in Samsung SDI, Korea, in 1991-1997, and a Research Fellow at Liquid Crystal Institute, Kent State University, USA, in 2001-2003, respectively. Since 2004, he has been with the department of electronics engineering, Dong-A University, Busan, Korea. His research interests include display devices.

Bong-Soon Kang received the B.S. degree in electronics engineering from Yonsei University, Seoul, Korea, in 1985, and the M.S. degree in electrical engineering. from University of Pennsylvania, Pennsylvania, USA, in 1987, and the Ph.D. degree in electrical and computer engineering from Drexel University, Philadelphia, USA, in 1990. From Dec. 1989 to Feb. 1999, he had worked as a senior staff researcher at Samsung Electronics Co. Ltd., Kihung, Korea. Since March 1999, he has been with the department of electronics engineering, Dong-A University, Busan, Korea. He is the director of the Multimedia Research Center of the university. His research interests include image processing, hardware architecture designs, and wireless communications. He was honored as a 2007 winner of the Chester Sall Award for the 1st place best paper in the IEEE Transactions on Consumer Electronics on Jan. 2009.