This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2015.2407773, IEEE Transactions on Circuits and Systems II: Express Briefs

Fully Pipelined Low-Cost and High-Quality Color Demosaicking VLSI Design for Real-Time Video Applications

Shih-Lun Chen, *IEEE Member*, and Huan-Rui Chang

***Abstract*—This paper presents a fully pipelined color demosaicking design. To improve the quality of reconstructed images, a linear deviation compensation scheme was created to increase the correlation between the interpolated and neighboring pixels. Furthermore, immediately interpolated green color pixels are first to be used in hardware-oriented color demosaicking algorithms, which efficiently promoted the quality of the reconstructed image. A boundary detector and boundary mirror machine were added to improve the quality of pixels located in boundaries. In addition, a hardware sharing technique was used to reduce the hardware costs of three interpolators. The VLSI architecture in this work contains only 4.97 K gate counts and the core area is 60,229 um^2 synthesized by using 0.18-um CMOS process. The operating frequency of this work is 200 MHz by consuming 4.76 mW. Compared with the previous low-complexity designs, this work has the benefits in terms of low cost, low power consumption, and high performance.**

***Index Terms*—Boundary detection, boundary mirror, CFA, digital camera, linear deviation compensation, and VLSI.**

I. Introduction

NOWADAYS, digital cameras are widely used in many fields. The most well-known is portable consumer electronics such as smart phone, tablet PC, digital video, digital camera, notebook, etc. The image sensor is a very important component in a digital camera, which can be roughly classified into charge-coupled device (CCD) and complementary metal oxide semiconductor (CMOS). A color filter array (CFA) technique is an efficient and compact method to obtain a multispectral image both on CCD and CMOS image sensors. Each pixel in a CFA image contains only one color.

The most widely used format of color filter arrays in the modern electronic products is Bayer color filter array [1] as shown in Fig. 1, in which each pixel contains one of red, green and blue colors only. Hence, the cost of image sensors and memory demand can be greatly reduced by the CFA technique. However, since two-third color information is missed after using color filter array, it is necessary to interpolate missing values backing to the CFA image to restructure a full color image.

Recently, some high-performance and high-quality color demosaicking and interpolation algorithms have been proposed. An orientation-free edge strength filter is proposed in [2]. It utilized edge information to avoid averaging non-correlated color differences and improved demosaicking performance successfully. A stochastic estimation approach to adaptive interpolation of color filter array was proposed in [3]. A gradient edge detection masks and adaptive heterogeneity-projection for demosaicking the color filter array was proposed by Chung *et al.* [4]. In order to avoid artifacts such as zipper effect, blur, and color spots, a self-similarity color demosaicking algorithm [5] was exploited to interpolate miss colors, which can be treated more generally as a graph-based regularization to the image. Besides, another novel graph-based regularization framework was presented in [6], in which a weight matrix was built to measure similarity and a static Laplacian was used to solve a variational problem. The artificial effect can be alleviated efficiently. Recently, a voting-based directional demosaicking algorithm is proposed in [7]. It uses a voting-base edge direction detection and a directional weighted methods to improve the quality of reconstructed images. Although the color demosaicking and interpolation algorithms as mentioned achieved high-quality, it is hard to realize these algorithms by using the VLSI technique for real-time video applications.

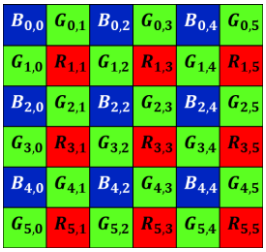

Fig. 1 Bayer color filter array (CFA).

Therefore, several hardware-oriented color demosaicking algorithms [8]-[12] were proposed for the VLSI implementation. Hsia *et al.* [8] proposed an edge-direction weighting and local gain approach based color demosaicking algorithm. In this design, the chip area cannot decrease obviously due to the dividers and multipliers were used. Hence, a cost-efficient method was proposed in [9] by using only simple operations such as addition, subtraction and shifter. A low-complexity algorithm based on the edge information and inter-channel correlations methods was presented in [10], in which a resource sharing and pipeline scheduling techniques were used to reduce hardware cost and improve throughput. A cost-effective and pipelined architecture based on a modified color difference spaces method by using direction determination was proposed in [11]. Recently, an adaptive edge-enhanced method including an anisotropic weighting model and filter-based compensator was presented in [12].

In this paper, a novel cost-efficient and high-performance hardware-oriented color demosaicking algorithm is proposed

Manuscript received July 22, 2014. This work was supported by the National Science Council, R.O.C, under Grant numbers of NSC-102-2221-E-033-063, NSC-102-2622-E-033-009-CC2, NSC-102-2815-C-033-022-E, MOST-103-2221-E-033-070, MOST-103-2218-E-033-004, and MOST-103-2622-E-033-001-CC2, CYCU-EECS-10301, and the National Chip Implementation Center, Taiwan. This paper was recommended by Associate Editor Yong-Bin Kim.

S. L. Chen and H. R. Chang are with the Department of Electronic Engineering, Chung Yuan Christian University, Chung Li City 320, Taiwan (ROC) (e-mail: chrischen@cycu.edu.tw).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2015.2407773, IEEE Transactions on Circuits and Systems II: Express Briefs

for VLSI implementation. Since the demosaicking equation for different colors have some similar parts, they can be integrated into a shared VLSI architecture by using the hardware sharing technique. Furthermore, in order to meet the demand of real-time video applications, the pipeline scheduling technique is used to realize this design.

II. Proposed Color Demosaicking Algorithm

The proposed color demosaicking algorithm consists of a novel linear deviation compensation, immediately interpolated green color information, boundary detection, and boundary mirror models as illustrated in Fig. 2. The boundary detector and mirror models were added to detect the boundary information and provide mirror pixels for green and red-blue color interpolations, which can efficiently improve the quality of interpolated pixels located in boundary. In addition, the linear deviation compensation and immediately interpolated green color pixels G' are used to improve the quality of the interpolated red and blue color pixels. The details of green and red-blue color interpolation models are described in the following subsections.

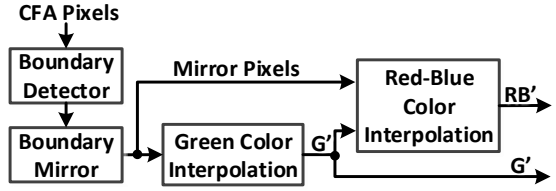

Fig. 2. Block diagram of the proposed color demosaicking algorithm.

A. Green Color Interpolation

When the locations of target pixels in original CFA image are red and blue colors, the green color pixels should be demosaicked by the green color interpolation model as shown in Equation (1). Since there is more information in horizontal than vertical directions, in order to avoid the difference between different rows, the result of green color pixels *G'* captures the CFA pixels in the horizontal direction only. Moreover, it is easy to use the same architecture to be implemented with Equation (4) by the hardware sharing technique. The results of Equation (1) will be stored into the second register bank as immediately interpolated green color pixels for red-blue color interpolation.

$$G'_{i,j} = \frac{1}{2}\left(G_{i,j-1} + G_{i,j+1}\right) + \frac{1}{2}RB_{i,j} - \frac{1}{4}\left(RB_{i,j-2} + RB_{i,j+2}\right) \quad (1)$$

B. Red-Blue Color Interpolation

In contrast with traditional bilinear guided demosaicing method, the immediately interpolated green color pixel *G'* from Equation (1) is used to improve the quality of red-blue color interpolation. The immediately interpolated pixels are first to be used in hardware-oriented color demosaicking algorithm. Moreover, a novel linear deviation compensation is also used to promote the performance of red-blue color interpolation. The difference between *G'* and the average of surrounding green color pixels in original CFA image is used to compensate the interpolated red and blue pixels. Finally, the quality of the reconstructed red and blue color pixels can be significantly improved by using the immediately green color information and linear deviation compensation techniques. The details of red-blue color interpolation method are described in the following.

Equation (2) shows the calculation of the $RB'^{(BR)G}_{i,j}$ where the original CFA pixel is $B_{i,j}$ or $R_{i,j}$. The $R'_{i,j}$ and $B'_{i,j}$ can be calculated by the average of four neighboring pixels ($RB_{i-1,j-1}$, $RB_{i-1,j+1}$, $RB_{i+1,j-1}$, $RB_{i+1,j+1}$) and then compensated by green colors. The linear deviation compensation technique which uses the difference between $G'_{i,j}$ and eight surrounding original and interpolated green color pixels as presented in the latter part of equation (2). The quality of interpolated red and blue pixels can be improved efficiently by the proposed linear deviation compensation technique.

$$RB'^{(BR)G}_{i,j} = \frac{1}{4}\left(RB_{i-1,j-1} + RB_{i-1,j+1} + RB_{i+1,j-1} + RB_{i+1,j+1}\right) + G'_{i,j} - \frac{1}{8}[\left(G'_{i-1,j-1} + G'_{i-1,j+1} + G'_{i+1,j-1} + G'_{i+1,j+1}\right) + \left(G_{i-1,j} + G_{i,j-1} + G_{i,j+1} + G_{i+1,j}\right)] \quad (2)$$

Equation (3) shows the calculation of the $RB'^{(G)BR}_{i,j}$ where the original CFA pixel is $G_{i,j}$. Since the color of $RB'_{i,j}$ is the same as $RB_{i-1,j}$ and $RB_{i+1,j}$ in the vertical direction, the average of $RB_{i-1,j}$ and $RB_{i+1,j}$ is the most information to interpolate the pixel $RB'_{i,j}$. Moreover, the function of the linear deviation compensation technique used in Equation (3) is the same as that used in Equation (2). Hence, the hardware cost can be reduced greatly by sharing the linear deviation compensation with Equation (2), which can improve the quality of the interpolated pixels both in Equations (2) and (3) by the same linear deviation compensator efficiently.

$$RB'^{(G)BR}_{i,j} = \frac{1}{2}\left(RB_{i-1,j} + RB_{i+1,j}\right) + G_{i,j} - \frac{1}{8}[(G'_{i-1,j} + G'_{i,j-1} + G'_{i,j+1} + G'_{i+1,j}) + (G_{i-1,j-1} + G_{i-1,j+1} + G_{i+1,j-1} + G_{i+1,j+1})] \quad (3)$$

The $RB'^{(G)RB}_{i,j}$ is the result of $RB_{i,j}$ where the original CFA color is $G_{i,j}$, and the colors of $RB_{i,j-1}$ and $RB_{i,j+1}$ are the same as $RB'_{i,j}$. In order to achieve high-quality and low-complexity, the reconstruction process for $RB'^{(G)RB}_{i,j}$ is only capturing the CFA pixels in the horizontal direction. Hence, the $R'_{i,j}$ and $B'_{i,j}$ can be obtained by using the CFA pixels in the horizontal direction as shown in Equation (4).

$$RB'^{(G)RB}_{i,j} = \frac{1}{2}\left(RB_{i,j-1} + RB_{i,j+1}\right) + \frac{1}{2}G_{i,j} - \frac{1}{4}\left(G_{i,j-2} + G_{i,j+2}\right) \quad (4)$$

The eight surrounding values of G' information used in Equations (2) and (3) can be obtained from Equation (1). Since the interpolation equations have the characteristics of high similarity, the hardware sharing technique can be used to reduce hardware cost efficiently.

III. VLSI Architecture

Fig. 3 shows a five-stage pipelined architecture of the proposed color demosaicking algorithm. It consists of a register bank 1, a boundary detector, a boundary mirror machine, three hardware sharing M1, M2, and M3, a register bank 2 and a controller. All parameters in the proposed algorithm are designed as 2, 4 and 8. Hence, the hardware cost can be greatly reduced by using the shifters rather than dividers and multipliers. The following subsections described the details of each block.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2015.2407773, IEEE Transactions on Circuits and Systems II: Express Briefs

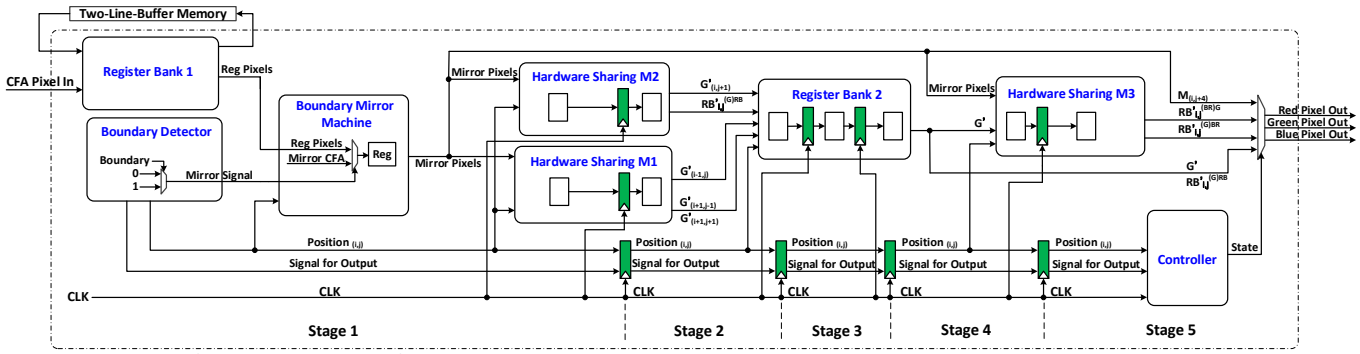

Fig. 3. Block diagram of the VLSI architecture for the proposed color demosaicking design.

A. *Register Banks*

The register bank 1 is composed of sixteen shift registers to provide the sixteen original CFA pixels as input for boundary mirror machine. By connecting with a two-line-buffer memory as shown in Fig. 3, the proposed color demosaicking design achieves pixel in and pixel out for real-time video applications. The register bank 2 consists of six registers as shown in Fig. 4. Since the red-blue color interpolation uses the information of immediately interpolated green color pixels G', it is necessary to store the green color pixels of the upper, middle, and lower row as inputs for the hardware sharing M3 to perform the linear deviation compensation. Moreover, the register bank 2 also used to balance the arriving time of $RB'^{(G)RB}_{i,j}$ and $G'_{i,\ j+1}$ because the result of $RB'^{(G)RB}_{i,j}$ is one cycle later than $G'_{i,\ j+1}$.

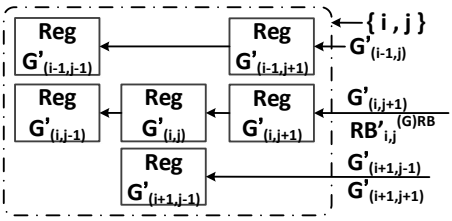

Fig. 4. Architecture of the register bank 2.

B. *Boundary Detector and Boundary Mirror Machine*

The boundary detector is designed to provide a flag signal to the boundary mirror machine to determine whether the location of the target pixel is within the boundary region or not. If the location is within the boundary range, the boundary mirror machine will provide the missed CFA values in the boundary region by a mirror technique. Moreover, six extra shift registers were added to store the information of the mirror CFA pixels for the hardware sharing M1 and M3.

C. *Hardware Sharing M1*

Fig. 5 shows the architecture of the hardware sharing M1. It can be realized by Equation (1) to obtain the results of $G'_{i-1,\ j}$, $G'_{i+1,\ j-1}$, and $G'_{i+1,\ j+1}$. The hardware sharing M1 consists of three adders, a substractor, three shifters, and three registers. The three registers were added to develop the pipeline scheduling. In order to save the hardware cost, the hardware sharing M1 is shared to interpolate the green color pixels of the upper and lower rows. It is used to obtain the result of $G'_{i-1,\ j}$ when the color of the target pixel $M_{i,\ j}$ in CFA image is green. Otherwise, when the color of the target pixel $M_{i,\ j}$ in CFA image is red and blue, it is used to calculate the result of $G'_{i+1,\ j-1}$ and $G'_{i+1,\ j+1}$ respectively. By adding six multiplexers to select different input signals, the hardware sharing M1 can obtain $G'_{i-1,\ j}$, $G'_{i+1,\ j-1}$, and $G'_{i+1,\ j+1}$ successfully. The results of $G'_{i-1,\ j}$, $G'_{i+1,\ j-1}$, and $G'_{i+1,\ j+1}$ will be stored in the register bank 2 as inputs for the hardware sharing M3. By using the hardware sharing technique, the computing resource of green color interpolator achieves reduction of 50 % adders, subtractors, and shifters.

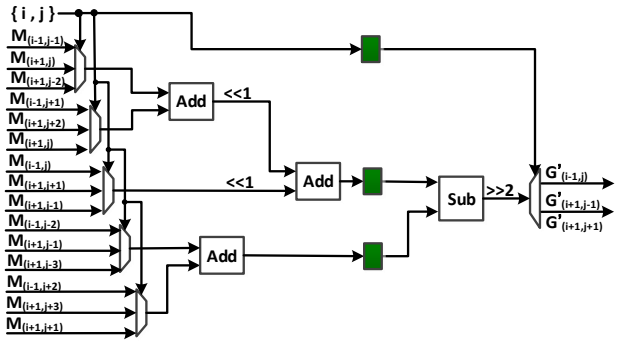

Fig. 5. Architecture of the hardware sharing M1.

D. *Hardware Sharing M2*

Fig. 6 shows the architecture of the hardware sharing M2. It is used to calculate the results of $G'_{i,j+1}$ and $RB'^{(G)RB}_{i,j}$ and it can be realized by equations (1) and (4). The hardware sharing M2 consists of three adders, a substractor, three shifters, and three registers. The three registers were also designed to develop the pipeline scheduling. Moreover, since the results of $G'_{i,j+1}$ and $RB'^{(G)RB}_{i,j}$ are complementary, only a multiplexer was added to achieve hardware sharing for $G'_{i,j+1}$ and $RB'^{(G)RB}_{i,j}$. The results of $G'_{i,\ j+1}$ were stored in the register bank 2 as inputs for the hardware sharing M3. There are 50% adders, subtractors, and shifters were reduced by using the hardware sharing technique.

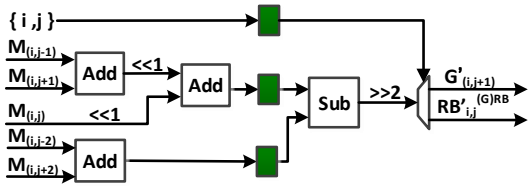

Fig. 6. Architecture of the hardware sharing M2.

E. *Hardware Sharing M3*

Fig. 7 shows the architecture of the hardware sharing M3. It is designed to obtain the results of $RB'^{(BR)G}_{i,j}$ and $RB'^{(G)BR}_{i,j}$ and can be implemented by equations (2) and (3). It consists of eleven adders, a subtractor, three shifters, and five registers. The five registers were added to develop the pipeline scheduling. The hardware sharing M3 captures the original green color pixels in the CFA image from boundary mirror

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2015.2407773, IEEE Transactions on Circuits and Systems II: Express Briefs

machine and the interpolated green color pixels by hardware sharing M1 and M2 as inputs to calculate the results of $RB'^{(BR)G}_{i,j}$ and $RB'^{(G)BR}_{i,j}$. The difference between the interpolated green color pixels G' and original green color pixels in the CFA image can be considered as the linear deviation with the ideal value at that point. The linear deviation technique is used to compensate the results of $RB'^{(BR)G}_{i,j}$ and $RB'^{(G)BR}_{i,j}$, which greatly improves the quality of the reconstructed image by using the proposed linear deviation compensation technique. Since the equations (2) and (3) are similar, the red-blue interpolator can be implemented by hardware sharing technique to significantly save 45% adders, 50% subtractors, and 50% shifters.

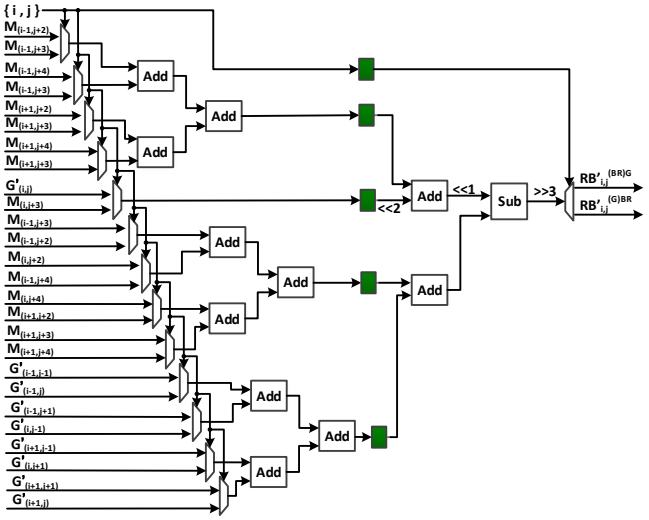

Fig. 7. Architecture of the hardware sharing M3.

IV. Simulation Results and Chip Implementation

In order to obtain the performance of the proposed color demosaicking algorithm, the MATLAB distribution was used to simulate the quality of reconstructed images by the hardware-oriented algorithms of LHCI [8], CDSP [9], ACDS [10], EECP [12], and this work. A twenty-four photographs of kodak data-set which are wildly used in image process field were selected as testing patterns. Each image in Kodak data-set is at the resolution of 768×512 with 24 bit-per-pixel. Furthermore, four popular formulas peak signal-to-noise ratio (PSNR), structural similarity (SSIM) [13], color peak signal-to-noise ratio (CPSNR), and S-CIELAB △E* metric [4], were used to evaluate the qualities of the interpolated images as shown in the following Equation (5)-(11). The value of PSNR can be calculated by

$$MSE = \frac{1}{M*N} * \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [P_{(i,j)} - P'_{(i,j)}]^2 \quad (5)$$

$$PSNR = 10 * log_{10}(\frac{MAX_P{}^2}{MSE}) \quad (6)$$

where he M and N are the width and height of the image, respectively. The $P_{(i,j)}$ and $P'_{(i,j)}$ are the pixel value of the original and reconstructed images with the position at the i^{th} row and j^{th} column, respectively. The MAX_p is the maximum value of pixels in a picture. Since each color in the testing image consists of 8-bit per pixel, the value of MAX_P is equal to 255. The *MSE* is the mean squared error between the original golden image and reconstructed images.

The CPSNR formula is similar to the PSNR formula as shown in equations (7) and (8).

$$MSE_{avg} = \frac{1}{3} * (\frac{1}{M*N}) * \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sum_{c \in C} [P^c_{(i,j)} - P'^c_{(i,j)}]^2 \quad (7)$$

$$\text{CPSNR} = 10 * log_{10}(\frac{MAX_P{}^2}{MSE_{avg}}) \quad (8)$$

where the C consists of red, green, and blue color components. The $P^c_{(i,j)}$ and $P'^c_{(i,j)}$ are the values of the pixel $P_{(i,\ j)}$ in the original and reconstructed image, respectively. The higher PSNR and CPSNR values denoted that the reconstructed image has more similarity to the original image. In addition, indexes called structural similarity (SSIM) [13] and S-CIELAB △E* metric [4] were used to quantify the reconstructed images if it is closer to the human eyes or not. The concept of the SSIM index is from human visual system and it can be calculated as

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x + \mu_y + C_1},\ c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x + \sigma_y + C_2},\ s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (9)$$

$$SSIM = [l(x,y)] * [c(x,y)] * [s(x,y)] \quad (10)$$

where σ_x and σ_y are the variance, the σ_{xy} is the covariance, and the μ_x and μ_y are the average values of the original and reconstructed images. Since the value of $\sigma_x^2 + \sigma_y^2$ is very close to zero, three parameters c_1, c_2, and c_3 are added to avoid instability. The $l(x,y)$, $c(x,y)$, and $s(x,y)$ are the luminance, contrast ratio, and structure between the image x and y, respectively. The other index S-CIELAB △E* metric can be calculated by

$$\triangle E^* = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \{\sqrt{[L_{(i,j)} - L\prime_{(i,j)}]^2 + [a_{(i,j)} - a\prime_{(i,j)}]^2 + [b_{(i,j)} - b\prime_{(i,j)}]^2}\}}{M*N} \quad (11)$$

where the M and N are resolutions of the image, and L is the luminance value. Variable a is the color between green and red, and variable b is the color between yellow and blue. The $L_{(i,\ j)}$, $a_{(i,\ j)}$, and $b_{(i,\ j)}$ are the CIELAB color component values of the pixel $P_{(i,\ j)}$ in the original image. The $L'_{(i,\ j)}$, $a'_{(i,\ j)}$, and $b'_{(i,\ j)}$ are the CIELAB color component values of the pixel $P_{(i,j)}$ in the reconstructed image. The less value of S-CIELAB △E* metric means the better interpolated image quality is achieved.

Table I lists the comparison of the four formula values and computing time for the previous designs with this work. The average CPSNR value in this work is 34.584 dB, which is better than 30.71 dB, 32.21 dB, 32.8 dB, and 34.377 dB in previous design [8], [9], [10], and [12], respectively. Compared with the previous designs [8-10, 12], this work improved the average values of PSNR, CPSNR, and S-CIELAB △E* metric by over 0.06 dB, 0.207 dB, and 0.053, respectively. In addition, an experimental result of computing time is added for comparing time complexity, which is the average time to demosaick per image of Kodak database by a computer with Intel core i7-2630QM when operated at 2.6 GHz. The results showed that this work had lower time complexity than the previous works. In addition, Fig. 8 shows

TABLE I
COMPARISONS OF THE AVERAGE IMAGE QUALITY AND COMPUTING TIME FOR VARIOUS COLOR DEMOSAICKING ALGORITHMS WITH THIS WORK

	LHCI [8]	CDSP [9]	ACDS [10]	EECP [12]	This work
PSNR (dB)	31.10	32.63	33.21	34.81	34.87
CPSNR (dB)	30.710	32.206	32.799	34.377	34.584
SSIM	0.9718	0.9793	0.9798	0.9834	0.9829
S-CIELAB	8.0529	7.0040	6.5283	5.5701	5.5172
Executing Time (s)	4.0976	2.8444	7.7248	7.9299	2.6067

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2015.2407773, IEEE Transactions on Circuits and Systems II: Express Briefs

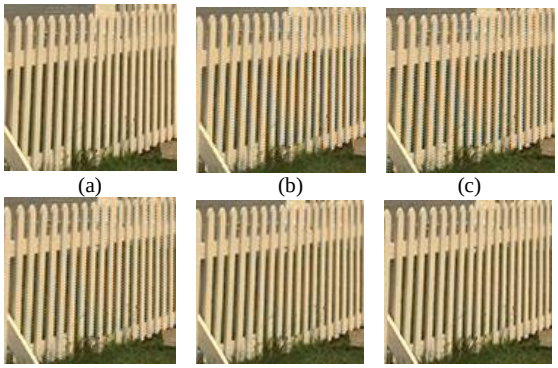

Fig. 8. Cropped region on a part of the interpolated *lighthouse* image for (a) original (b) LHCI (c) CDSP (d) ACDS (e) EECP (f) this work.

TABLE II
COMPARISONS OF COMPUTING RESOURCE FOR PREVIOUS DESIGNS WITH THIS WORK

	[8]	[9]	[10]	[12]	This Work	
					Without Hardware Sharing	With Hardware Sharing
Division	2	0	0	0	0	0
Multiplication	3	0	0	0	0	0
Absolute	1	0	2	6	0	0
Addition	6	20	17	30	32	17
Subtraction	1	4	5	10	6	3
Shift	0	10	13	15	18	9

a cropped region of the interpolated lighthouse image by the algorithms of LHCI [8], CDSP [9], ACDS [10], EECP [12], and this work for comparing the visual quality of high frequency part and zipper effect of the interpolated image.

Table II lists the comparisons of computing resource for the previous designs with this work. The computing resource in this work contains only 17 additions, 3 subtractions, and 9 shifts. By using the hardware sharing technique, the 46.8 % adders, 50 % subtractors, and 50 % shifters were successfully reduced in this work. This work can be realized by adders, subtractors, and shifters only without any dividers and multipliers. The silicon area of a divider or multiplier is much greater than an adder, subtractor, and shifter. Moreover, compare with previous design [12], this work achieved reduction of 43.3 % adders, 70 % subtractors, and 40% shifters and without using any absoluter. This work was implemented by using a hardware description language and an EDA tool of Design Vision (Synopsys) was used to synthesize this design with the library of TSMC 0.18-μm CMOS process. The NAND-equivalent gate count in this work is only 4.97 K and its power consumption is 4.76-mW when operates at 200MHz. The core area is 60,229-μm^2, in which the width and length are 243.85-μm and 246.99- μm, respectively.

Table III lists the comparisons of the previous low-complexity color demosaicking designs with this work. Compared with the previous designs, this work improved the average CPSNR by over 3.874 dB, 2.378 dB, 1.785 dB, and 0.207 dB than the previous designs LHCI [8], CDSP [9], ACDS [10], and EECP [12], respectively. Moreover, this work also saved over 50.3 %, 80.8 %, 11.2 %, and 4.4 % gate counts than the previous designs LHCI [8], CDSP [9], ACDS [10], and EECP [12], respectively. The memory requirement in this work is only a two-line-buffer memory, it is much less than 1 frame in [9]. Compared with the previous low-complexity designs, this work not only improved the quality of the interpolated images but also reduced the hardware cost and memory requirement. It provided an efficient color demosaicking VLSI design for real-time video applications.

TABLE III
COMPARISONS OF PREVIOUS LOW-COMPLEXITY COLOR DEMOSAICKING DESIGNS WITH THIS WORK

	LHCI [8]	CDSP [9]	ACDS [10]	EECP [12]	This Work
CPSNR (dB)	30.710	32.206	32.799	34.377	34.584
Process	0.35 μm	0.35 μm	0.18 μm	0.18 μm	0.18 μm
Gate Counts	10 K	26 K	5.6 K	5.2 K	4.97 K
Frequency (Hz)	40 M	50 M	200M	200 M	200 M
Power	200 mW	56 mW	150 mW	4.66 mW	4.76 mW
Core area (μm^2)	7 M	5 M	670 K	64.2 K	60 K
Memory	2 lines	1 frame	2 lines	2 lines	2 lines
Memory area (μm^2)	78.4 K	15.8 M	20.7 K	20.7 K	20.7 K
Throughput (pixel/s)	40 M	50 M	200 M	200 M	200 M
Quality	VGA	VGA	HD	HD	HD
Normalized Area	2.01	5.23	1.13	1.05	1

V. Conclusion

In this paper, a cost-efficient and high-performance color demosaicking VLSI design based on hardware sharing and pipeline scheduling techniques is proposed for real-time video applications. A linear deviation compensation, immediately interpolated green color pixels, a boundary detector and a boundary mirror machine are used to improve the quality of the reconstructed image.

References

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

[2] I. Pekkucuksen, and Y. Altunbasak, "Edge strength filter based color filter array interpolation," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 393–397, Jan. 2012.

[3] H. A. Chang, and H. H. Chen, "Stochastic color interpolation for digital cameras," *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 17, no. 8, pp. 964-973, Aug. 2007.

[4] K. L. Chung, W. J. Yang, W. M. Yan, and C. C. Wang, "Demosaicing of Color Filter Array Captured Images Using Gradient Edge Detection Masks and Adaptive Heterogeneity-Projection," *IEEE Trans. Image Processing*, vol. 17, no. 12, pp. 2356–2367, Dec. 2008.

[5] A. Buades, B. Coll, J.-M. Morel, and C. Sbert, "Self-similarity driven color demosaicking," *IEEE Trans. Image Process.*, vol. 18, no. 6, pp. 1192–1202, Jun. 2009.

[6] C. Hu, L. Cheng, and Y. M. Lu, "Graph-based regularization for color image demosaicking," *in Proc. IEEE Int. Conf. Image Processing (ICIP)*, Oct. 2012, pp. 2769–2772.

[7] X. Chen, G. Jeon, and J. Jeong, "Voting-based directional interpolation method and its application to still color image demosaicking," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 24, no. 2, pp. 255-262, Feb. 2014.

[8] S. C. Hsia, M. H. Chen, and P. S. Tsai, "VLSI implementation of low-power high-quality color interpolation processor for CCD camera," *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, Vol. 14, no. 4, pp. 361-369, Apr. 2006.

[9] S. C. Hsia, and P. S. Tsai, "VLSI implementation of camera digital signal processor for document projection system," *in Proc. IEEE Int. Conf. Signal Processing System (ICSPS)*, Jul. 2010, pp. 657–660.

[10] Y. H. Shiau, P. Y. Chen, and C. W. Chang, "An area-efficient color demosaicking scheme for VLSI architecture," *International Journal of Innovative Computing, Information and Control*, Vol.7, No.4, pp.1739-1752, Apr. 2011.

[11] H. Chen and Y. Cheng, "VLSI implementation of color interpolation in color difference spaces," *in Proc. IEEE Int. Conf. Circuits and Systems (ISCAS)*, May 2012, pp. 1680–1683.

[12] S. L. Chen and E. D. Ma, "VLSI Implementation of an Adaptive Edge-Enhanced Color Interpolation Processor for Real-Time Video Applications," *IEEE Transaction on Circuits and Systems for Video Technology*, 2014. (Accepted)

[13] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.