EFFECTIVE COLOUR DEMOSAICKING WITH DIRECTIONAL FILTERING AND WEIGHTING IN DIGITAL IMAGE PIPELINING

M.Anitha Pakiyaraj[1] | S.Vaisali[2]
[1](Assistant Professor, M.P. Nachimuthu M. Jaganathan Engineering College, Erode, India, nishamera2k12@gmail.com)
[2](ECE, M.P. Nachimuthu M. Jaganathan Engineering College, Erode, India, vaisali26@gmail.com)

Abstract*—Digital cameras usually use a single sensor covered with a colour filter array which samples only one colour at the location of each pixel. Restoring a full-colour image by the demosaicking technique is a key task in the digital imaging pipeline. This paper proposes a novel demosaicking approach based on the existing directional filtering and weighting techniques. The contributions of this paper are two-fold. Firstly, we analyse the advantages and limitations of the the existing directional filtering and weighting techniques, and improve the two existing techniques in order to suppress the common demosaicking artifacts. Digital cameras usually use a single sensor covered with a colour filter array which samples only one colour at the location of each pixel. Restoring a full-colour image by the demosaicking technique is a key task in the digital imaging pipeline. This paper proposes a novel demosaicking approach based on the existing directional filtering and weighting techniques. The contributions of this paper are two-fold. Firstly, we analyse the advantages and limitations of the the existing directional filtering and weighting techniques, and improve the two existing techniques in order to suppress the common demosaicking artifacts. Secondly, we give a new estimated scheme for the reconstruction of the colour components. Experimental results show that the proposed method outperforms the recent six state-of-the-art demosaicking methods in terms of both subjective and objective measures. Secondly, we give a new estimated scheme for the reconstruction of the colour components. Experimental results show that the proposed method outperforms the recent six state-of-the-art demosaicking methods in terms of both subjective and objective measures.*

***Keywords*—Boundary mirror, Boundary detector, Directional filtering and weighting, Colour Filter Array**

1. INTRODUCTION

Digital cameras have become ubiquitous during the last decade. In a digital camera, a colour image is often produced using red (R), green (G), and blue (B) three colour samples at each pixel location. In order to reduce cost and size, a lot of cameras use a single sensor covered with a colour filter array (CFA). The CFA allows only one colour to be measured at each pixel. The missing other two colour values at each pixel must be estimated. This estimation process is known as demosaicking.

Consumer-level digital cameras were introduced in mid-1990s; in about a decade, the digital camera market has grown rapidly to exceed film camera sales. Today, there are point-and-shoot cameras with over 8 million pixels; professional digital SLR cameras with more than 12 million pixels are also available. Resolution, light sensitivity, and dynamic range of the sensors have been significantly improved such that image quality of digital cameras has become comparable to that of film cameras. During an image capture process, a digital camera performs various processing including auto-focus, white balance adjustment, color interpolation, color correction, compression and more. An important component of the imaging pipeline is color filter array (CFA) interpolation - i.e., to recover an full-resolution image from its CFA data.

Among many CFA patterns, the most commonly used is the Bayer. The Bayer pattern measures the green image on a quincunx grid (half of the image resolution) and the red and blue images on rectangular grids (quarter of the image resolution). The green channel is measured at a higher sampling rate than the other two because the peak sensitivity of the human visual system (HVS) lies in the medium wavelengths, corresponding to the green portion of the spectrum.

Recently, many CFA patterns have been found to exist. The most common pattern is the Bayer CFA. Here we only consider the Bayer pattern illustrated in Figure in which the G values are sampled on a quincunx grid, while the R and B values are sampled on rectangular grids.

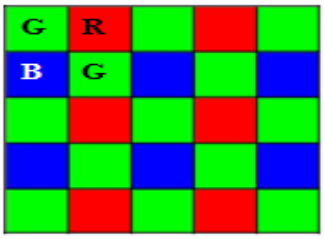


Fig. 1: Bayer colour filter

The image is measured at a higher sampling rate because the peak sensitivity of the human visual system lies in the medium wavelengths, corresponding to the G portion of the spectrum.

So far there have been an immense number of demosaicking algorithms available. In particular, some sophisticated algorithms have been published for the last several years. These algorithms have generally notable advantages over the early algorithms. The early best known algorithm is perhaps bilinear interpolation (BI) which

interpolates the missing colours of each pixel by averaging the colour values of nearby pixels. BI is very simple and has also a good effect in the smooth regions of an image, but it suffers heavily from visual artifacts around the edges of an image. Hamilton and Adams proposed an adaptive colour plane interpolation method (ACPI), which considers the first and second order derivatives in order to predict the direction of an edge. The reconstruction of the G plane is added with the second-order correction terms from the R and B planes, while the reconstructions of the R and B planes are added with the second-order correction terms from the G plane. ACPI has had an important influence on the later demosaicking algorithms.

2. PROPOSED COLOR DEMOSAICKING ALGORITHM

The main objective of this study is to explore how to efficiently improve and combine the existing directional filtering and weighting schemes in order to alleviate the common demosaicking artifacts. Because DFPD almost removes the grid errors in the G plane, our interpolation in the G plane is based on DFPD, and our main improvement to DFPD consists in the interpolations of the missing G values on the weak edges by the weighted average of two orthogonal directional G values; analogously, because EECI can commendably suppress false colours, our interpolations in the R and B planes have a similar weighting scheme to EECI, and our main improvement to EECI consists in calculating weights on the colour difference planes.

The proposed algorithm includes two steps: initial step and refinement step. In the initial step, the initial estimates of the R, G and B planes are computed, while in the refinement step, the G plane can be further refined using the initial R and B planes and vice versa. The principle of the proposed algorithm will be further illustrated in the following.

A. Interpolation of the Green Plane

We have seen in Section II that the edge-directed DFPD almost eliminates the zipper artifacts in the example of Fig. 2. However, in the textured regions, the detected direction of an edge tends to be incorrect and the inordinate false colours are caused (Fig. 3). Based on the observations, we consider improving the interpolation of the G plane in DFPD. For an explicit horizontal or vertical edge, we estimate the missing G values using DFPD. Otherwise the G values are estimated using the weighted average of two orthogonal directional G values, where the choice of weights is dependent on the strength of the horizontal and vertical directional edges across the missing pixel. The interpolation of the G plane comprises two steps: directional interpolation step and decision step.

1) Directional interpolation step: This step interpolates the missing G values using the ACPI interpolator in horizontal and vertical directions respectively. Estimating the missing G value at an R pixel will be considered [Fig. 4]. The interpolation at a B pixel can be carried out similarly. Let's find G5 at R5.

$$\begin{cases} G_5^H = (G_4 + G_6)/2 + (R_5 - R_3 + R_5 - R_7)/4 \\ G_5^V = (G_2 + G_8)/2 + (R_5 - R_1 + R_5 - R_9)/4 \end{cases}$$

2) Decision step: Once the G plane has been interpolated along both horizontal and vertical directions, two G images will have been produced. It is required to select a better direction from the two directions for every pixel. Therefore, two colour difference images are computed as:

$$C_H(i,j) = \begin{cases} R_{i,j} - G_{i,j}^H, & \text{if } (i,j) \text{ is a red location} \\ B_{i,j} - G_{i,j}^H, & \text{if } (i,j) \text{ is a blue location} \end{cases}$$

$$C_V(i,j) = \begin{cases} R_{i,j} - G_{i,j}^V, & \text{if } (i,j) \text{ is a red location} \\ B_{i,j} - G_{i,j}^V, & \text{if } (i,j) \text{ is a blue location} \end{cases}$$

where i and j indicate the row and the column of the pixel (i; j). Next, the gradients of CH and CV are calculated as DH(i; j) = jCH(i; j)CH(i; j +2)j and DV (i; j) = jCV (i; j)CV (i+2; j)j.

Subsequently, two classifiers H(i; j) and V (i; j) are defined as the sum of the gradients DH and DV in a 5x5 neighbourhood centered at pixel (i; j). They give an estimate of the local variation of the colour differences along the horizontal and vertical directions respectively, and can be used to estimate the directions of the edges. For all R and B pixels, the missing G values are estimated using the following criterion:

$$\begin{cases} \text{if } (1+\delta_V(i,j)) / (1+\delta_H(i,j)) > T \\ \text{then} \\ \quad G_{i,j} = G_{i,j}^H \\ \text{else if } (1+\delta_H(i,j)) / (1+\delta_V(i,j)) > T \\ \quad G_{i,j} = G_{i,j}^V \\ \text{else} \\ \quad G_{i,j} = \left(w_1 * G_{i,j}^H + w_2 * G_{i,j}^V\right) / (w_1 + w_2) \\ \text{end} \end{cases}$$

where 1 is added to the classifiers to avoid division by zero. T is a threshold and needs to be carefully chosen because the ratio between the two classifiers, suggests the strength of the horizontal or vertical edge. The greater the value of T is, the larger the smooth region divided into is. The choice on T will be discussed in Section IV. The above criterion indicates that if the missing G value Gi;j is on a horizontal strong edge (or on a vertical strong edge), then Gi;j = GH i;j (or Gi;j = GV i;j), or else Gi;j is in a textured or weak edge regions and takes a weighted average of GH i;j and GV i;j , whose contributions to Gi;j should be proportional to the strength of the horizontal and vertical edges.

Because the strength of a directional edge is inversely proportional to the directional gradient, the weights w1 and w2 can be considered as the reciprocal of the directional gradients. We compute the horizontal and vertical gradients using the exact same method as described. The calculations of the weights w1 and w2 have two cases (Fig. 1) where the location (i; j) of the missing G pixel is an R pixel or a B pixel. Here calculating w1 and w2 at an R pixel will be considered [Fig. 4], in which the location 5 corresponds to

the location (i; j) of the missing G pixel. The calculation at a B pixel can be carried out similarly.

The weights are computed as follows:

$$\begin{cases} w_1 = 1/(1 + |G_6 - G_4| + |2R_5 - R_3 - R_7|) \\ w_2 = 1/(1 + |G_8 - G_2| + |2R_5 - R_1 - R_9|) \end{cases}$$

where 1 is also added to the denominators to avoid division by zero.

B. Interpolations of the Red and Blue Planes

Once the G plane is fully populated, it will be used to assist the subsequent interpolations of the R and B planes because the estimated G values can be considered known. It has also been seen that the directional filtering of the G plane can significantly suppress zipper artifacts, and the methods based on weighting such as EECI can better suppress false colours than the edge-directed ones (Fig. 3). Here we consider that the interpolations of the missing R and B pixel values use a weighted method. The proposed algorithm has a similar weighting scheme to EECI, but we calculate weights on the colour difference planes, while EECI calculates weights on the mosaic image.

The justification is that colour difference images are smoother than primary colour planes. It is well known that demosaicking algorithms tend to provide satisfactory results within the smooth regions of an image. Our weights also have the different forms from the EECI's. They can detect the wider edges than the EECI's, because our weights are calculated in a 7x7 neighbourhood while the EECI's are done in a 5x5 neighbourhood. In the following, only the interpolation of the R pixel value at a B pixel location will be considered. The interpolation of the B pixel value at an R pixel location can be carried out analogously.

There are three different cases for the interpolation of an R pixel value (Fig.5). The missing R pixel values at the B positions will first be estimated because every B pixel has four adjacent R pixels in the diagonal directions. With the populated G plane, the colour difference KR = G R can be computed at all R pixel positions. Then every missing R pixel can be estimated by the four adjacent colour different values. For example, to estimate the missing R9. R9 = G9 KR9 in which

$$KR_9 = \frac{w_6 * KR_6 + w_7 * KR_7 + w_{11} * KR_{11} + w_{12} * KR_{12}}{w_6 + w_7 + w_{11} + w_{12}}$$

where KRi = Gi Ri(i = 6; 7; 11; 12) and the weights are computed as:

$$\begin{cases} w_6 = \frac{1}{1+|KR_{12}-KR_6|+c|KR_6-KR_1|} \\ w_7 = \frac{1}{1+|KR_{11}-KR_7|+c|KR_7-KR_4|} \\ w_{11} = \frac{1}{1+|KR_7-KR_{11}|+c|KR_{11}-KR_{14}|} \\ w_{12} = \frac{1}{1+|K_6-KR_{12}|+c|KR_{12}-KR_{17}|} \end{cases}$$

where c is a constant factor adjusting the weighting effect, whose choice will be discussed in Section IV. The principle computing the weights similar to the one.

Next the missing R pixel value at a G position will be estimated. Every G pixel has four available adjacent R pixels in horizontal and vertical directions after the missing R pixels are interpolated at the B pixel positions [Figs. 5(a) and (b)]. There into two R pixels in the horizontal or vertical direction are estimated. For example, to estimate the missing R5, then

R5 = G5-K

Similarly,

Where $$KR_i = G_i - R_i (i = 2, 4, 6, 8)$$

C. Refinement

After the initial interpolation step, the three colour planes are fully populated and their estimates can be further refined. With the aid of the populated R and B planes, the G plane can be refined. Then the refined G plane is used to refine the R and B planes again. Only the estimated sample values will be refined and the original CFA-sampled colour values at each pixel are not altered. In addition, our refinement is only performed once. Because the refinement is to exploit the spectral correlation between colour planes, if the correlation is overused, the demosaicking results will become worse.

1) Refinement of the green plane: Refining the G value at an R pixel will be considered. The refinement at a B pixel can be carried out similarly. Let's refine G5 at R5 [Fig. 6(a)]. G5 = R5 +KR5, where KR5 is computed and its weights are computed.

2) Refinements of the red and blue planes: With the refined G plane, the R and B planes can be further refined. Since refining the B plane is similar to refining the R plane, only refining the R plane will be considered. The R values which need refining are the ones at the G or B sampling positions. With the initial populated R values, both G and B pixel positions have four adjacent R values in the horizontal and vertical directions. For every G pixel position, two of the adjacent R values are from the original CFA samples and the other two ones are estimated values. For every B pixel position, four adjacent R values are all estimated values. The refinements of the R values at the G and B positions are all performed.

3. The RAW Image Editing Workflow

In order to work with and display in MATLAB images originating from sensor data, we must take into account the aforementioned nature of the raw data. The workflow depicted in Figure is a first order approximation of how to get a 'correct' displayable output image from the raw sensor data. Section will cover how to implement this in MATLAB, but this can also be considered the general approach using any programming language.

Fig. 2 workflow for raw image processing.

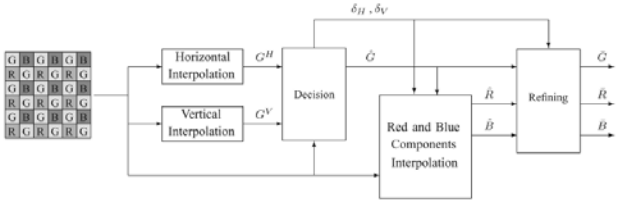


Fig. 3 Complete scheme of the proposed algorithm

4. Determination of Threshold T and Constant Factor c

The proposed method needs to determine the threshold T in Equation and the constant factor c in Equations. The parameter T is very important because it balances strong edges and smooth regions. The larger T value results in the larger smooth regions and less strong edges, whereas the smaller T value results in the smaller smooth regions and more strong edges. The improper T value can incur additional demosaicking artifacts. The parameter c can slightly improve weighting effect. Unfortunately, the two parameters cannot be easily determined because they depend on the varying scene structures of images. We derive T and c through training. Fourteen of twenty-four standard Kodak colour images were used as training samples (the other ten Kodak colour images), They were sampled according to Bayer CFA to form the mosaic images and were then used as the input for the demosaicking algorithms. A full three-colour representation of colour images was reconstructed using the proposed method with the different T.

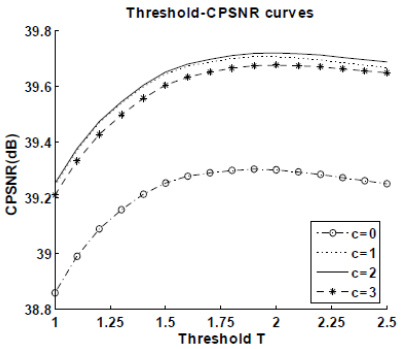


Fig. 4 Training results of the parameters of test images

5. EVALUATION OF ALGORTITHM

To evaluate the performance of our proposed algorithm, its demosaicked results were compared with those of the recent six state-of-the-art demosaicking algorithms—AP, DFPD, EECI, adaptive heterogeneity-projection (AHP), directional linear minimum mean square-error estimation (DLMMSE), and regularization approaches to demosaicking (RAD). The source or executable codes of all algorithms are achieved directly from the original authors. The popular Kodak set of test images are commonly used to benchmark demosaicking performance. However, some researchers and have noted that the Kodak test images have higher spectral correlations than typical colour images. The demosaicking algorithms overemphasizing spectral correlations can produce highly visible colour artifacts. In our experiments, the used test images are shown in Fig. 8 including the ten Kodak images of size 512_768 pixels (Images 1-10) and the two other images.

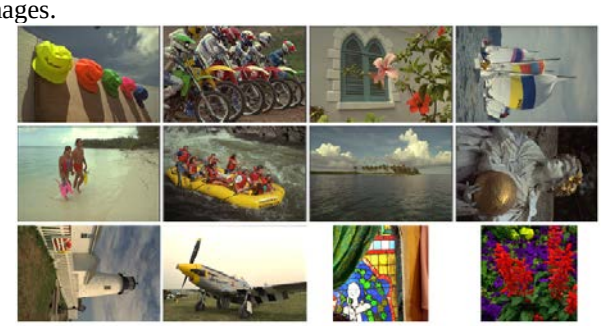


Fig.5 Set of test image

Two metrics colour peak signal-to-noise ratio (CPSNR) and S-CIELAB (The MATLAB code for the S-CIELAB metric is also available were used as the objective measures for comparing the demosaicking algorithms. All errors were calculated excluding the border of six pixels around the image to reduce the boundary effects. The threshold T in (2) was set to 1:5 and the constant factor c in was set to 2. The CPSNR results are summarized in Table I. The highest CPSNR in each row is highlighted in bold. The results in terms of average S-CIELAB metric are given in Table II. The minimum S-CIELAB is also highlighted in bold. The S-CIELAB metric extends the CIELAB metric to colour images. Like the CIELAB metric, the S-CIELAB metric is a “perceptual colour fidelity” metric. It measures how accurate the reproduction of a colour is to the original when viewed by a human observer.

Objective measures are not often consistent with the subjective visual quality. The visual quality of the reconstructed images for different demosaicking algorithms should be evaluated by the visual inspection. The first example is Image 9 with the high spectral correlation. Fig above shows a cropped region of Image 9 and the corresponding demosaicked results with different techniques. This example includes a picket fence, with high spatial frequency along the fence which often suffers from zippering artifacts. It can be seen that DFPD, AHP, DLMMSE, and the proposed algorithm obviously outperform the other two ones, but a more careful examination can find that the proposed algorithm presents the least aliasing artifacts. The second example is Image 12 with the weak spectral correlation. Fig above is part of Image 12. On this specific example, EECI and the proposed algorithm are evidently superior to the others. The quality of the demosaicked images using proposed algorithm and EECI seems to be indiscernible.

6. SIMULATION RESULTS AND COMPARISONS

6.1Existing Results

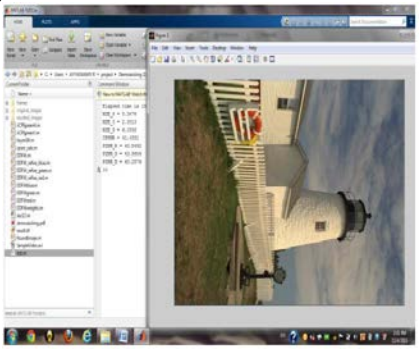


Fig.6 Simulated results for Existing work using MATLAB

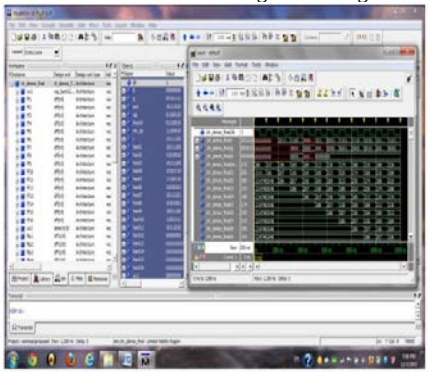
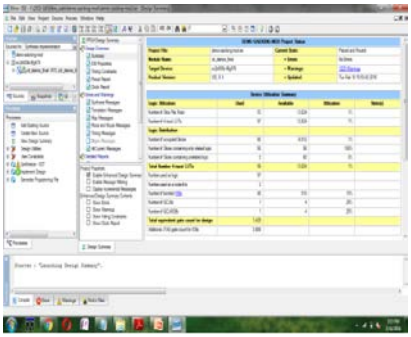


Fig.7 Simulated results for Existing work using Modelsim



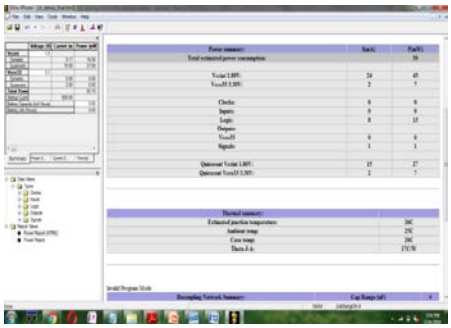


Fig.8 Simulated results for Existing technique using Xilinx

6.2 Proposed Results

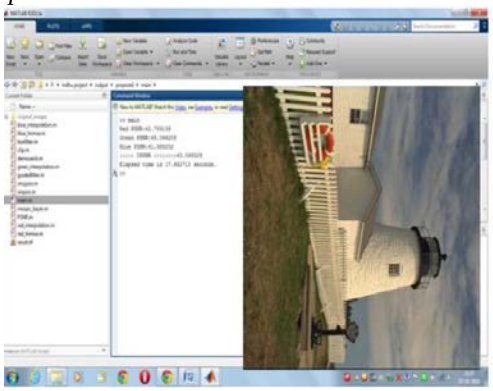


Fig.9 Simulated results for proposed work using MATLAB

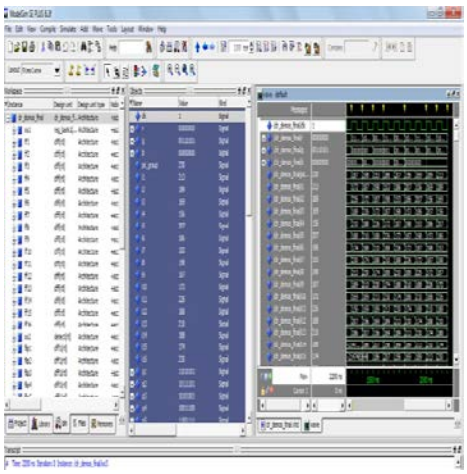
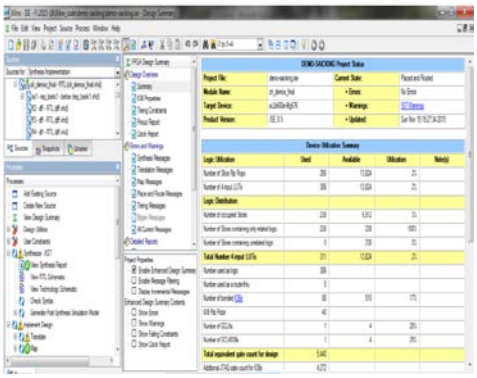


Fig.10 Simulated results for proposed work using Modelsim



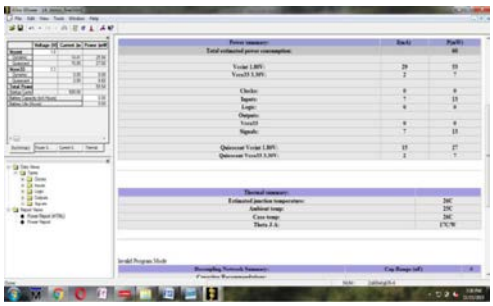


Fig.11 Simulated results for proposed technique using Xilinx

6.3 Comparison Table

TABLE I COMPARISONS OF AVERAGE IMAGE QUALITY WITH EXISTING ALGORITHMS AND PROPOSED WORK

Parameters	Existing	Proposed
CPSNR	41.4331	43.0993
PSNR_R	40.8492	42.7591
PSNR_G	43.9956	45.0462
PSNR_B	40.2876	41.9882

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 41.4 dB, which is better than 30.71, 32.21, 32.8, and 33.377 dB in the previous designs. Compared with the previous designs, this work improved the average values of PSNR, CPSNR respectively.

Table 6.2 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.

TABLE 6.2 COMPARISONS OF COMPUTING RESOURCE FOR PREVIOUS DESIGNS WITH PROPOSED WORK

	Existing Method	Proposed Work
CPSNR	41.4331	43.0993
Gate Counts	5440	1429
Frequency	13.417ns	12.135ns
Power	60mW	50mW

By using the hardware sharing technique, 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 a multiplier is much greater than that of an adder, a subtractor, and a shifter. Moreover, compared with a previous design, 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 Modelsim tool to provide best results since it easily connects matlab to modelsim. By using the Xilinx tool the total gate counts used in this work is only 5440 gate counts and its power consumption is only 60mW. When this work is implemented in an EDA tool of Design Vision (Synopsys) 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 200 MHz. The core area is 60229 μm2, in which the width, length are 243.85 and 246.99-μm, respectively.

Moreover, this work also saved over 50.3%, 80.8%, 11.2%, and 4.4% gate counts than the previous designs LHCI, CDSP, ACDS, and EECP, respectively. The memory requirement in this work is only a two-line-buffer memory device; it is much less than 1 frame. 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. Thus the simulated results and comparison with previous techniques have been demonstrated as mentioned above.

7. CONCLUSION

The proposed work concludes that a cost efficient and high performance color demosaicking VLSI design based on hardware sharing and pipeline scheduling techniques has been proposed for real time video applications. It uses the residual interpolation method to improve the quality of the reconstructed image. Also it reduces the hardware cost and power by using SPST technique.

ACKNOWLEDGMENT

The authors acknowledge the contributions of the students, faculty of M.P.Nachimuthu M.Jaganathan Engineering College for helping in the design of check circuitry, and for tool support. The authors also thank the anonymous reviewers for their thoughtful comments that helped to improve this paper. The authors would like to thank the anonymous reviewers for their constructive critique from which this paper greatly benefited.

REFERENCES

[1] B. E. Bayer, “Color imaging array,” U.S. Patent 3 971 065, Jul. 1976.
[2] H. Chen and Y. Cheng, “VLSI implementation of color interpolation in color difference spaces,” in Proc. IEEE ISCAS, May 2012, pp. 1680–1683.
[3] H. A. Chang and H. H. Chen, “Stochastic color interpolation for digital cameras,” IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 8, pp. 964–973, Aug. 2007.
[4] Kuan-Hung Chen, “A Spurious-Power Suppression Technique for Multimedia/DSP Applications”Member, IEEE, and Yuan-Sun Chu, Member, IEEE
[5] 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.
[6] 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, pp. 361–369, Apr. 2006.
[7] S. L. Chen and E. D. Ma, “VLSI implementation of an adaptive edge-enhanced color interpolation processor for real-time video applications,” IEEE Trans. Circuits Syst. Video Technol., vol. 24, no. 1991, pp. 1982–345, Nov. 2014.
[8] S. C. Hsia and P. S. Tsai, “VLSI implementation of camera digital signal processor for document projection system,” in Proc. IEEE ICSPS, Jul. 2010, pp. 657–660.
[9] Y. H. Shiau, P. Y. Chen, and C. W. Chang, “An area-efficient color demosaicking scheme for VLSI architecture,” Int. J. Innov. Comput., Inf. Control, vol. 7, no. 4, pp. 1739–1752, Apr. 2011.
[10] Daisuke Kiku, Yusuke Monno, Masayuki Tanaka, and Masatoshi Okutomi “Residual Interpolation for Color Image Demosaicking”
[11] 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.

BIOGRAPHY

M.Anitha completed her M.E in Mahendra Engineering College, Completed B.E Electronics and Communication Engineering in Kongu Engineering College,Perundurai and have 9 years of teaching experience. Now working as a assistant professor in ECE department in M.P.Nachimuthu M.Jaganathan Engineering College, Erode,India.

Vaisali S.Spursuing M.E VLSI Design in M.P.Nachimuthu M.Jaganathan Engineering College, Erode, Tamilnadu, India and She completed his B.E Electronics and Communication Engineering in T.R.P. Engineering college, Trichy,Tamilnadu, India.His area of interest includes VLSI Design and Verification and Digital electronics.