DEMOSAICING OF IMAGES FOR BAYER CFA USING PROJECTION ALGORITHM

R.Niruban[1], T.Sree Renga Raja[2], R.Deepa[3]

[1] Assistant Professor, Electronics and Communcation Engineeering, Anand Institute of Higher Technoloy,TN ,India
[2] Associate Professor,Electrical and Electronics Department, Anna university,Thiruchirapalli,,TamilNadu,India
[3] Assistant professorComputer scienc and engineering,Dr.K.Vasudevan college of engineering and Technology,TamilNadu,India,

Abstract

Typical consumer digital camera sense only one out of three components per image pixel because of increase in size and cost of sensor used in camera. An effective demosaicing is presented to restore the missing pixels of image captured from single sensor cameras. To eliminate most of color artifacts in edge region, Edge based demosaicing algorithm is to interpolate missing green sample followed by interpolate red and blue samples. Many demosaicing algorithms find edges in horizontal and vertical directions, which are not suitable for other directions. Before using the algorithm Gaussian filter is used for edge enhancement and smoothing of image. This proposed algorithm will be compared with other existing algorithms using PSNR measure.

Key Words: *Demosaiking , Gaussian filtering, Iterative demosaicing, Refining*

---***---

1. INTRODUCTION

Digital devices include modern computers, mobile phones, laptop and other electronic devices are inbuilt with digital cameras, because of small in size and cost-effective. Many peoples are choosing devices like mobile phone to capture image instead of using high digital cameras, the reason is low cost and also more convenient.

Single sensor used in this camera is CCD (or) CMOS sensor and the filter used is a color filter array. CMOS sensors required relatively low power where as CCD sensors [4-7] respond in low light condition also. Therefore both this sensors deliver high quality image at the same time consumer friendly. Filter used with single sensor to capture the image called CFA(Color Filter Array) but it samples only one color in an image. There are different CFA patterns but the commonly used one is a Bayer pattern which composed of 50% of green, 25% of red and 25% of blue. It will sample more green values because of human eye is more sensitive to green compare to red and blue. Remaining color samples missing in each pixel in bayer pattern are interpolated by using the method called demosaicing (or) color filter array interpolation.

There are many demosaicing algorithms used for interpolating missing color samples. First one is the bilinear interpolation which interpolates red value in non red pixel and interpolates blue value in non blue pixel, similarly for green by using two or four adjacent red and blue pixels for compute red and blue samples respectively. Bilinear interpolation does not interpolate in edge region and cause zipper effect in edge regions. Cubic interpolation [2] is similar to [1] and at the same time to interpolate R,G,B it take the four pixels that are closest to the diagonal corners and average their value to produce the middle pixel. Bicubic interpolation, in contrast takes not only the four closest diagonal pixels, but their closest pixels as well, for a total of 16 pixels. This method is efficient but introduces large errors in edge regions.

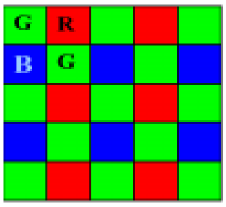

Fig 1[11]: Bayer Pattern

The above described methods are non adaptive algorithms. Now we discuss about the adaptive methods. In non linear method also called adaptive method green sample is interpolated first where red and green is interpolated by using green.

In smooth hue transition method, cross correlation takes place, it removes the false color during the demosaicing process, but it suffers from color artifacts. It uses constant color difference method between red, green and blue pixels.
Another non linear method used is edge gradient based interpolation method. Gradient based method [13] is nothing but extensive of adaptive color plane method (ACPI)[17]. In ACPI we consider only two directions but in this method two more directions are considered to calculate gradient of missing G samples. Weight is determined for each direction using known R or B and G values.

In this section, a method to determine green by using four direction gradient is developed and also implement the algorithm which defines feasible sets based on obtained colors in each pixel and experience the inter channel correlation between the channels. Our algorithm is compared with various existing demosaicing algorithm by using PSNR measurement. Section 2 presents the preprocessing process. Section 3 declares the benefits and purpose of this algorithm. The comparison of other existing algorithm with our algorithm is given in section 4. Conclusion is provided in section 5.

2. PREPROCESSING

The two step pre processing procedure is used in this paper. Image must be suitable for further processing and matlab environment so preprocessing process takes place. Here Gaussian filter is used for preprocessing which enhances the image quality and also used for image restoration. Gaussian filtering is used to remove noise and weight gives high significance to the pixels near edge.

2.1 Gaussian Filter: The impulse response function of Gaussian filter is the function of Gaussian filter. Mathematically, Gaussian filters alternating the input value by using convolution with impulse response. The Gaussian filter is a continuous and is not a discrete function. The cut off frequency Fc of the Gaussian filter is calculated as the ratio between square root of variance and sample rate Fs.

The Gaussian filter is defined by the equation

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{x^2}{2\sigma^2}} \qquad (1)$$

The impulse response function of 1D Gaussian filter is given as

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-\sigma^2 u^2}{2}} \qquad (2)$$

In this preprocessing setup, CFA image is filtered and enhanced by Gaussian filter results in error reduction smoothening of image and it is passed through Gaussian filter for further processing.

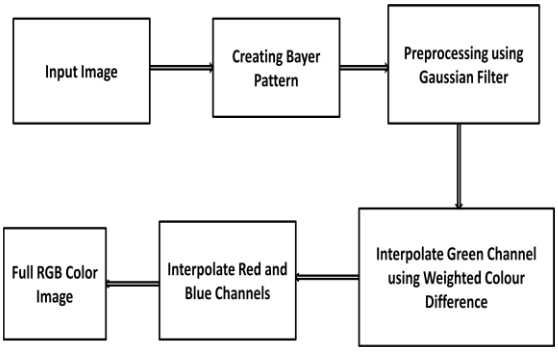

Fig 2: BLOCK DIAGRAM

3. DEMOSAICING USING WEIGHTED EDGE METHOD

In natural images, there is a high correlation between red, green and blue channels. The correlation between these three samples is same in both texture and edge region. While using camera having CFA in which luminance called green is sampled at higher rate than chrominance (red, blue pixels). Therefore performance of red and blue pixels is little poor as compare to the green and at the same time aliasing effect is high in red and blue samples. This paper suggests a new demosaicing algorithm with feasible sets used for remove color artifacts and aliasing in high frequency regions include edges. It sense that interpolated missing red and blue is observed and reconstruct them by using cross channel correlation which is projecting onto feasible sets.

Before applying this constraint set, decompose the channel by using filter bank structure. In filter bank $H_0(z)$ is a low pass filter and $H_1(z)$ is a high pass filter. Section 3(A) defines how the missing green in every pixel is interpolated using edge gradient based interpolation method. In section 3(B) declares interpolation of R and B using cubic interpolation. Section C defines feasible sets used by the proposed algorithm.

4. DEMOSAICING USING WEIGHTED EDGE METHOD

Image consists of three color planes. Red, Blue, and Green plane. In our proposed algorithm, the missing color plane is interpolated using remaining color planes. Based on this method, missing green is determined first later missing red and blue color samples in each pixel is determined. Here original image pixel value is R,G,B and R^*,G^*,B^* are interpolated pixel value.

In order to find missing green in non green pixels, determine weight in four sides of required pixel then the color difference in four sides of pixel is determined by neighboring pixel value. The weight assign to color difference is used to estimate the missing green pixel. Finally calculate two color difference planes Kr and Kb. In first step, we have only to determine the gradient for the red and blue channel. The gradient is the green channel so we consider on green pixel that is actually around the red pixel. Likewise interpolate green pixel from the blue component. Let us take fig-3 to explain how to estimate G channel value at R33.Before performing interpolation for G channel determine the gradient for red and blue pixel.

R1	G12	R13	G14	R15
G21	B22	G23	B24	G25
R31	**G32**	**R33**	**G34**	**R35**
G41	B42	G43	B44	G45
R51	G52	R53	G54	R55

Fig -1: 5x5 Bayer pattern

As shown in Fig (3), to obtain the G33 value at the R33 pixel first to calculate the gradient along the four adjacent directions as follows

$$\alpha_{2,3} = |R_{1,3} - R_{3,3}| + |G_{2,3} - G_{4,3}|$$
$$\alpha_{4,3} = |R_{5,3} - R_{3,3}| + |G_{2,3} - G_{4,3}|$$
$$\alpha_{3,2} = |R_{3,1} - R_{3,3}| + |G_{3,2} - G_{3,4}|$$
$$\alpha_{3,4} = |R_{3,5} - R_{3,3}| + |G_{3,2} - G_{3,4}| \quad (1)$$

The Four adjacent color differences is calculated as follows

$$K_{2,3} = G_{2,3} - 1/2\,(R_{1,3} + R_{3,3})$$
$$K_{4,3} = G_{4,3} - 1/2\,(R_{5,3} + R_{3,3})$$
$$K_{3,2} = G_{3,2} - 1/2\,(R_{3,1} + R_{3,3})$$
$$K_{3,4} = G_{3,4} - 1/2\,(R_{3,5} + R_{3,3}) \quad (2)$$

The weights as defined in equation (1) are then assigned to the four adjacent color difference values K, as defined In equation (2), The estimating k3,3 is calculated as follows

$$K1 = (K_{2,3})/(1+\alpha_{2,3})$$
$$K2 = (K_{4,3})/(1+\alpha_{4,3})$$
$$K3 = (K_{3,2})/(1+\alpha_{3,2})$$
$$K4 = (K_{3,4})/(1+\alpha_{3,4})$$

$$K_{3,3} = (K1+K2+K3+K4)/((1+\alpha_{2,3})+(1+\alpha_{4,3})+(1+\alpha_{3,2})+(1+\alpha_{3,4}))$$

Then the Missing Green pixel value in red CFA components is calculated as follows

$$G^*_{3,3} = R_{3,3} + K_{3,3} \quad (3)$$

Similarly the green pixel value is calculated in blue cfa components.

5. INTERPOLATING MISSING COMPONENT AT GREEN

After interpolating all green color planes of the image, interpolate missing red and blue samples in all green position. For interpolate red and blue at green position consider also the green components which is interpolated from red and blue. Figure.4 (a) shows the green CFA sample located in the red green row and blue green column to interpolate red and blue horizontally and vertically located pixels.

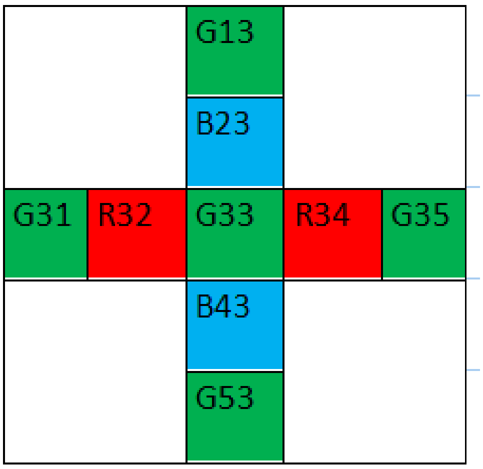

Fig 4(a): Interpolate missing R in G

$$R^*_{3,3} = G_{3,3} + [R_{3,2} - G^*_{3,2} + R_{3,4} - G^*_{3,4}]/2$$
$$B^*_{3,3} = G_{3,3} + [B_{2,3} - G^*_{2,3} + B_{4,3} - G^*_{4,3}]/2$$

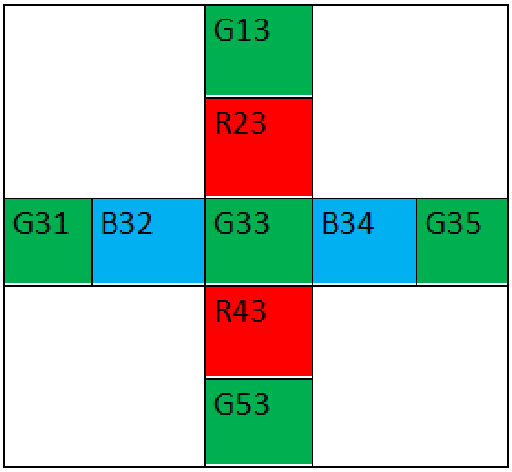

Fig 4(b): Interpolate missing B in G

Figure.4(b) shows the green CFA sample located in the red green column and blue green row to interpolate red and blue vertically and horizontally located pixels.

$$R^*_{3,3} = G_{3,3} + [R_{2,3} - G^*_{2,3} + R_{4,3} - G^*_{4,3}]/2$$
$$B^*_{3,3} = G_{3,3} + [B_{3,2} - G^*_{3,2} + B_{3,4} - G^*_{3,4}]/2$$

6. INTERPOLATE MISSING R(B) COMPONENT AT B(R) SAMPLING POSITION

Finally the missing red components at the blue sampling position are interpolated. Fig. 5, shows the possible cases where pixel of interest lies in the centre of a 5x5 window. Here red pixels are interpolated by using bilinear interpolation method over color difference. For the case in Fig.4 (b), the centre missing red sample, $R^*_{i,j}$, is interpolated by

$$R^*_{i,j} = G_{i,j} + \frac{1}{4}\sum_{m=\mp 1}\sum_{n=\mp 1}\left(R_{i+m,j+n} - G_{i+m,j+n}\right) \quad (4)$$

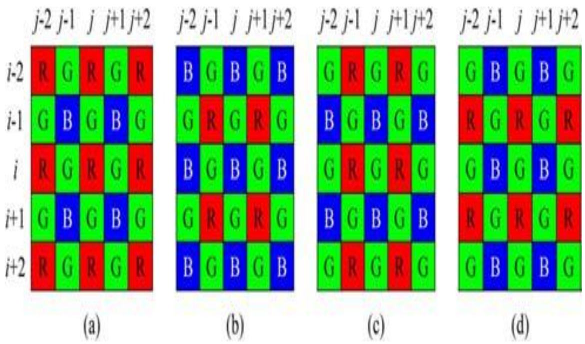

Fig 5[17]: Example of CFA configuration

As for the case in Fig. 5(a), the centre missing blue sample, $R^*_{i,j,}$ can also be obtained with (4) by replacing the red estimates with the corresponding blue estimates in (5). At last, the final full color image can be obtained.

The missing blue component in red is interpolated by

$$B^*_{i,j} = G_{i,j} + \frac{1}{4}\sum_{m=\mp 1}\sum_{n=\mp 1}\left(B_{i+m,j+n} - G_{i+m,j+n}\right) \quad (5)$$

7. FEASIBLE SETS

Plan a conceptual reconstruction process produce a solution which is accurate with the details appear from detected data or initial knowledge about the solution. For demosaicing approach in our method, two types of feasible sets, one based on the perceive data and other depend on cross-channel correlation. The missing color samples which are interpolated must be acting in same way with color samples already present in CFA of digital camera. We declared $D(t_1,t_2)$ as this observed data contains R,G,B placed according to Bayer pattern. R^*,G^*,B^* are set of pixel point.(t_1,t_2) which defines the pixel point have samples of red, green and blue samples respectively.

The equation for determined feasible set is as follows:

$$F_o = \{X(t_1,t_2) = D(t_1,t_2) \forall (t_1,t_2) \epsilon \wedge_s, (X) = R, G, B\} \quad (6)$$

Where X is a common symbol for interpolating missing color samples which may be R for red channel, G for Green and B for blue respectively.

The first observed data obtained using constraint set is not enough to define constraint sets of high frequency sub bands similar to luminance component. This section is used in obtaining feasible sets on red and blue channel where high frequency obtained is similar to the frequency of the green channel.

Reconstruction process is provide by decompose the feasible sets by using two filters $H_0(z)$ and $H_1(z)$. These two filters provide better reconstruction process with synthesis filter $S_0(z)$ and $S_1(z)$.

It can be written as $H_0(z)S_0(z) + H_1(z)S_1(z) = 1$

Let $h_0(.)$ &$h_1(.)$ denote the impulse response of $H_0(z)$ & $H_1(z)$. we can write four sub bands with two dimensional signal $X(t_1,t_2)$ as follows.

$W_1=[h_0(t_1)*h_0(t_2)]*X(t_1,t_2)$
$W_2=[h_1(t_1)*h_0(t_2)]*X(t_1,t_2)$
$W_3=[h_0(t_1)*h_1(t_2)]*X(t_1,t_2)$
$W_4=[h_1(t_1)*h_1(t_2)]*X(t_1,t_2)$ (7)

By denoting $S_0(.)$ & $S_1(.)$ as the impulse response of $S_0(z)$ & $S_1(z)$ then the filtering operations on a two level dimensional signal as follows

$Y_1=[S_0(t_1)*S_0(t_2)]*X(t_1,t_2)$
$Y_2=[S_1(t_1)*S_0(t_2)]*X(t_1,t_2)$
$Y_3=[S_0(t_1)*S_1(t_2)]*X(t_1,t_2)$
$Y_4=[S_1(t_1)*S_1(t_2)]*X(t_1,t_2)$ (8)

Where Y_1,Y_2,Y_3,Y_4 are synthesis filter operators. This form a reconstruction filter bank with approximate sub bands W_1,W_2,W_3,W_4.

$X(t_1,t_2)=Y_1W_1+Y_2W_2+Y_3W_3+Y_4W_4$

Now we define detail feasible set F_d as follows.

$F_d=\{X(t_1,t_2):|(W_1)(t_1,t_2)-(W_k)(t_1,t_2)|\le T(t_1,t_2)$
for K=2,3,4 X=R,B

It enhances the high frequency component of red and blue channel to be similar to green channel. Here T is the threshold which defines the color channel correlation. If the color channels are highly correlated, threshold gets to be small. If the color channel is not correlated threshold become larger. In order to set T to zero, high correlation of color channel is required.

7.1 Projection operator: The projection P_0 for observation constraint set is obtained from the feasible set given in (6). From that equation, we can write P_0 onto "observation" feasible set as follows:

$$P_0[X(t_1,t_2)] = \begin{cases} D(t_1,t_2), \epsilon \wedge s \\ X(t_1,t_2), otherwise \end{cases}$$

The projection P_d for detail feasible set F_d is given as

$$P_d[x(t_1,t_2)] = Y_1(W_1)(t_1,t_2) + \sum_{k=2}^{4} Y_k(W_k)(t_1,t_2)$$

8. SIMULATION RESULTS

Simulation was carried out to estimate the performance of the proposed algorithm. The 24 digital color images shown in Fig. 5 were used to generate a set of testing images. The peak signal-to-noise ratio (PSNR) was used as a measure to express the performance of the demosaicing methods. In PSNR, particular PSNR for each color sample is defined as

$$CPSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right)$$

Where,

$$MSE = \sum_{i,j} \frac{\left(I_{org}(i,j) - I_r(i,j)\right)^2}{m*n}$$

I_{org} -represent the original image of size $m*n$

I_r-represent the reconstructed image of size $m*n$

Images are numbered from 1 to 24 in the order of left-to-right and top-to-bottom was used in the experiments

Fig 6[11]: Images used for Simulation

Table- 1: Compare the proposed method with existing demosaicing algorithm

Image	Bilinear	Edge based	Subband synthesis	Wavelet with spatial refinement	Proposed method
1	30.773	31.2458	39.5224	40.6942	42.8064
2	36.6873	37.0425	39.5198	41.3527	45.6784
3	37.6775	38.0647	43.3218	43.9651	47.7373
4	37.4213	37.8047	41.4224	43.4641	46.6931
5	31.2157	31.8233	38.6369	40.5972	43.5113
6	32.1414	32.5964	40.255	41.9364	44.1239
7	37.1797	37.6152	42.1956	44.6154	46.7256
8	28.2879	28.7457	36.0158	38.9305	42.0271
9	36.4896	36.9292	43.2609	45.2354	47.4915
10	36.3952	36.8716	43.5337	45.2596	47.8490
11	33.6645	34.1372	41.006	42.9933	44.0718
12	36.9695	37.3912	43.5063	46.4256	49.6349
13	28.6113	29.1087	37.2893	37.3747	41.9504
14	33.469	33.9611	36.7377	38.7378	44.0018
15	35.4486	35.8213	39.8739	42.0606	46.1802
16	35.2366	35.6156	43.2864	44.6285	44.5141
17	36.681	37.2463	43.6912	44.4829	46.8972
18	32.562	33.0611	38.9911	40.0813	43.8988
19	32.7829	33.2797	40.5543	43.9703	45.6252
20	34.6422	34.9845	41.2988	42.7135	47.6793
21	33.0627	33.5363	40.9836	41.8151	45.1638
22	34.9277	35.3907	39.492	40.9682	45.3023
23	38.3544	38.7374	42.7698	43.9694	48.5216
24	31.411	31.8746	37.6036	38.0094	45.2865

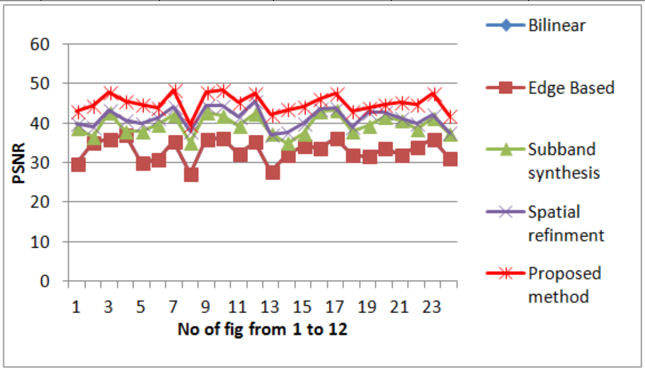

Fig 7 PSNR value comparisons of red channel with existing method

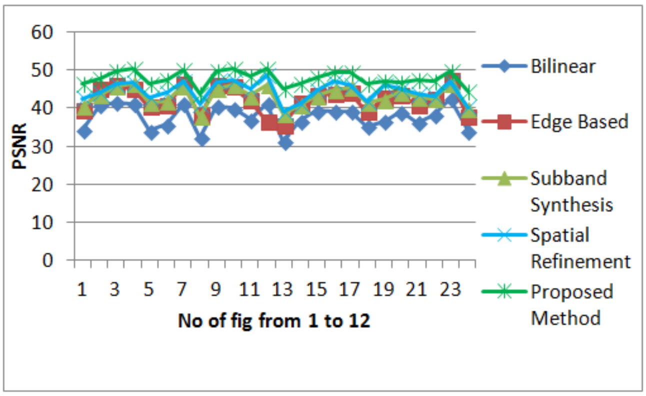

Fig 8: PSNR value comparisons of green channel with existing method

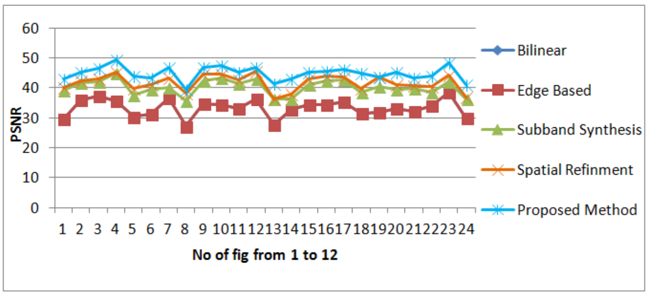

Fig 9: PSNR value comparisons of blue channel with existing demosaicing algorithms

Table -2: Compare the CPSNR value of proposed method with existing method

Img	Bilinear			Edge based			Subband synthesis			Wavelet with spatial refinement			Proposed method		
	R	G	B	R	G	B	R	G	B	R	G	B	R	G	B
1	29.741	34.286	29.678	29.741	39.442	29.678	38.881	40.591	39.272	39.811	42.617	40.154	43.1701	46.5192	43.0099
2	35.022	40.797	36.100	35.022	44.854	36.100	36.524	43.533	41.665	39.147	44.034	42.327	44.5201	47.8272	45.1250
3	35.942	41.591	37.252	35.942	45.912	37.252	42.825	45.853	42.113	43.217	46.289	43.078	47.7906	49.7287	46.5177
4	37.106	41.126	35.663	37.106	45.051	35.663	37.928	46.396	44.927	40.751	46.799	45.211	45.4878	50.4066	49.4103
5	30.126	34.082	30.430	30.126	40.240	30.430	37.853	41.500	37.547	39.959	42.717	39.705	44.6659	46.5055	43.8606
6	30.803	35.699	31.355	30.803	40.606	31.355	39.616	41.921	39.609	41.342	44.180	40.943	43.9950	47.5484	43.3552
7	35.514	41.092	36.662	35.514	46.389	36.662	42.159	45.984	40.226	44.156	47.043	43.420	48.4452	50.0155	46.7127
8	27.104	32.181	27.220	27.104	37.959	27.220	34.998	37.820	35.697	37.959	41.183	38.307	39.6434	43.6341	39.5488
9	36.102	40.438	34.717	36.102	45.917	34.717	42.785	45.225	42.298	44.552	46.919	44.624	47.7729	49.7786	46.7624
10	36.272	40.017	34.528	36.272	45.509	34.528	41.980	46.187	43.416	44.351	47.456	44.598	48.4588	50.3640	47.3625
11	32.269	37.035	33.016	32.269	41.880	33.016	39.313	43.338	41.279	41.738	45.071	42.801	45.2345	48.5037	45.2789
12	35.266	41.164	36.404	35.266	36.404	36.404	42.376	46.138	42.872	45.619	48.716	45.612	47.5609	50.4384	46.9416
13	27.650	31.312	27.764	27.650	35.342	27.764	37.443	38.584	36.173	37.128	39.053	36.363	42.1709	45.0075	41.2764
14	32.123	36.675	32.828	32.123	41.524	32.828	34.990	40.838	36.262	37.767	41.174	38.021	43.4756	46.3178	43.0135
15	34.216	39.315	34.439	34.216	43.306	34.439	37.339	43.381	41.055	39.927	44.670	42.936	44.2827	47.9899	45.1903
16	33.815	39.277	34.378	33.815	43.662	34.378	43.143	44.836	42.260	43.632	46.928	44.008	46.2854	49.4875	45.5253
17	36.205	39.248	35.440	36.205	43.949	35.440	43.318	45.433	42.754	43.964	46.035	43.786	47.4677	49.3932	46.1630
18	31.906	35.128	31.478	31.906	38.980	31.478	37.865	41.464	38.428	39.039	41.776	39.860	43.1096	46.4512	44.9765
19	31.592	36.521	31.772	31.592	42.845	31.772	39.476	42.313	40.337	42.768	45.944	43.773	43.9892	47.0844	43.6487
20	33.838	39.032	33.080	33.838	43.287	33.080	41.613	44.279	39.349	42.673	45.111	41.206	44.7645	46.6920	45.2220
21	31.892	36.279	32.221	31.892	40.838	32.221	40.720	42.780	39.923	41.446	43.773	40.762	45.1942	47.4017	43.4365
22	33.866	38.111	33.983	33.866	42.446	33.983	38.562	42.531	38.472	39.964	43.059	40.467	44.7391	47.0056	43.9862
23	36.114	42.556	38.598	36.114	47.324	38.598	41.226	46.550	42.145	42.076	47.010	44.159	47.5222	49.6431	48.4389
24	31.062	34.104	30.016	31.062	37.717	30.016	37.292	40.023	36.291	37.664	40.261	36.797	42.0336	44.3045	41.1282

From Table 2, it is acknowledged that bilinear interpolation method fails to detect the edge in images. In edge based method as shown in Table 3. having high CPSNR in G but not determine edges in R and B. As compare to edge based method, sub band synthesis having high PSNR but it cause color artifacts in images due to high frequency component. In spatial refinement process, zipper effect cause due to longer filters. However in proposed method, based on projection algorithm and gradient based interpolation method weight is determined for every pixel and refinement take place before and after interpolating of image. By comparing the performance of various algorithms, the proposed method gives better performance among the existing methods.

3. CONCLUSIONS

In this paper, Color Filter Array based on HVS and alternative demosaicing algorithm with enhanced border and edge was presented. It makes use of the color difference variance of the pixels located along the four directions in a local region to estimate the interpolation direction for interpolating the missing green samples. The high-performance arises from the introduction of a gradient based interpolation method. With them, the proposed algorithm has a good initial condition and can terminate algorithm early. The main advantage of the proposed interpolation is its computational efficiency for green. Simulation results show that the proposed method is able to process a subjectively and objectively best demosaicing results

REFERENCES

[1] Parulski, K. and K. E. Spaulding, 2002. Color image processing for digital cameras, in Digital Color Imaging Handbook, (eds.) G. Sharma, CRC Press, Boca Raton, FL., 2 :728-757.

[2] Dillon, P.L.P., D. M. Lewis, and F. G. Kaspar, 1978. Color imaging system using a single CCD area array, IEEE Journal of Solid-State Circuits, 13(1) : 28-33.

[3] Lule, T., S.Benthien, and H. Keller,et al., 2000. Sensitivity of CMOS based imagers and scaling perspectives, IEEE Transactions on Electron Devices, 47(11) : 2110-2122.

[4] Lukac, R. and K. N. Plataniotis, 2005. Data-adaptive filters for demosaicking: A framework, IEEE Transactions on Consumer Electronics,51(2):560-570.

[5] Technical Committee ISO/TC 42, 2001. Photography, Electronic still picture imaging - Removable memory, Part 2: Image data format - TIFF/EP, ISO 12234-2.

[6] Lukac, R. and K. N. Plataniotis, 2005. Fast video demosaicking solution for mobile phone imaging applications, IEEE Transactions on Consumer Electronics, 51(2): 675-681.

[7] Wu, X. and N. Zhang, 2004. Primary-consistent soft-decision color demosaicking for digital cameras, IEEE Transactions on Image Processing,13(9):1263-1274.

[8] Lukac, R., B. Smolka, K. Martin, K. N. Plataniotis, and A. N.Venetsanopulos, 2005. Vector filtering for color imaging, IEEE Signal Processing Magazine;22(1):74-86.

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

[10] Haddad, R.A., and A.N. Akansu, 1991. A Class of Fast Gaussian Binomial Filters for Speech and Image Processing, IEEE Transactions on Acoustics, Speech and Signal Processing,39:723-727.

[11] Gunturk, B.K. ,Y.Altunbasak and R. M. Mersereau, 2002. Color plane interpolation using alternating projections, IEEE Trans.on Image Process, 11(9): 997-1013.

[12] Parmar, M. and S.J. Reeves, 2004. A perceptually based design methodology for color filter arrays, in Proc.IEEE Int. Con. Acoustics, Speech,and Signal Processing (ICASSP'04), 3: 473- 476.

[13] Hirakawa, K. and T.W.Parks, 2005. Adaptive homogeneity–directed demosaicing algorithm, IEEE Trans .on Image Process, 141(39): 360-369.

[14] Alleysson, D., S. Süsstrunk, and J. Hérault,2005. Linear demosaicing inspired by the human visual system, IEEE Trans. Image Process, 14(4): 439-449.

[15] Haddad, R.A. and A.N. Akansu,1991. A Class of Fast Gaussian Binomial Filters for Speech and Image Processing, IEEE Transactions on Acoustics, Speech and Signal Processing,39: 723-727.

[16] Kakarala, R. and Z. Baharav, 2002. Adaptive demosaicing with the principal vector method, IEEE Trans. Consum. Electron.,48(11):932–937.

[17] king-Hong Chung,. and Yuk-Hee Chan, 2006. Color demosaicing using variance of color differences, IEEE Transactions on Image Processing, 15(10):2944-2955.