Research Journal of Applied Sciences, Engineering and Technology 7(15): 3049-3055, 2014
DOI:10.19026/rjaset.7.641
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: August 01, 2013 Accepted: August 23, 2013 Published: April 19, 2014

Research Article

Inter Channel Correlation based Demosaicking Algorithm for Enhanced Bayer Color Filter Array

[1]K. John Peter, [2]S. Arumugam and [3]K. Senthamarai Kannan
[1]VINS Christian College of Engineering, Anna University, Nagercoil, Tamil Nadu, India
[2]Nandha Engineering College, Erode, Tamil Nadu, India
[3]MS University, Tirunelveli, Tamil Nadu, India

Abstract: Demosaicking is a process of obtaining a full color image by interpolating the missing colors of an image captured from a digital still and video cameras that use a single-sensor array. In this study a new Color Filter Array (CFA) is proposed. Which is based on the actual weight of the Human Visual System. It is developed based on the sensitivity level of the human eye to red as 29.9%, green as 58.7% and blue as 11.4%. This study also provides an effective iterative demosaicing algorithm applying a weighted-edge interpolation to handle green pixels followed by a series of color difference interpolation to update red, blue and green pixels. Before applying demosaicking algorithm Gaussian filter is applied to remove noise of the sensor applied image and also enhance the image quality. Experimental results show that the proposed method performs much better than other latest demosaicing techniques in terms of image quality and PSNR value.

Keywords: Bayer's sampling, color difference, color interpolation, Gaussian filter, iterative demosaicing

INTRODUCTION

Color Filter Array (CFA) is one of the most distinctive hardware elements in a single-sensor imaging pipeline (Parulski and Spaulding, 2002). The CFA is placed on top of the monochrome image sensor, usually a Charge-Coupled Device (CCD) (Dillon *et al*., 1978) or Complementary Metal Oxide Semiconductor (CMOS) (Lule *et al*., 2000) sensor, to acquire the low-resolution color information of the image scene. Each sensor cell has its own spectrally selective filter and thus, the acquired CFA data constitutes a mosaic-like monochrome image (Lukac and Plataniotis, 2005b). Since the information about the arrangement of the color filters in the CFA is known from the camera manufacturers or it can be obtained using the Tagged Image File Format for Electronic Photography (TIFF-EP), the grayscale CFA image can be re-arranged as a low-resolution color image. This is the initial operation in the demosaicking process (Lukac and Plataniotis, 2005a; Wu and Zhang, 2004; Jayachandran and Dhanasekeran, 2012; Gunturk *et al*., 2005) which uses the concept of spectral interpolation to estimate the missing color components and to produce a full-color image (Lukac and Plataniotis, 2005b). The arrangement of the color filters in the CFA varies depending on the manufacturer (Bayer, 1976; Parmar and Reeves, 2004). Consumer electronic devices, such as various digital still and video cameras, image-enabled mobile phones and wireless Personal Digital Assistants (PDAs) thus naturally differ on the employed demosaicking solution. Different cost and implementation constraints are expected for a camera which stores the image in the CFA format and uses a companion personal computer to demosaick the acquired image data, than for a camera which directly produces the demosaicked image. Other construction differences may result from the intended application (e.g., Consumer photography, surveillance, astronomy). Among the various suggested CFAs in Fig. 1, the Bayer CFA pattern (Fig. 1a) is the most prevalent one, where *G* pixels occupy half of all and *R* and *B* pixels share the others. A representation of a full-color image needs all the information from the three colors at each pixel location. As a result, the missing two colors on each pixel location have to be interpolated back to get a full-color image. The process of interpolating the missing colors is called as demosaicing or color interpolation whose main objective aims to reconstruct the missing colors as closely to the original ones as possible while keeping the computational complexity as low as possible.

A demosaicing algorithm can be either heuristic or non-heuristic. A heuristic approach does not try to solve a mathematically defined optimization problem while a non-heuristic approach does. Most existing demosaicing algorithms are heuristic algorithms. It can be found that a number of heuristic algorithms were developed based on the framework of the Adaptive

Corresponding Author: K. John Peter, VINS Christian College of Engineering, Anna University, Nagercoil, Tamil Nadu, India

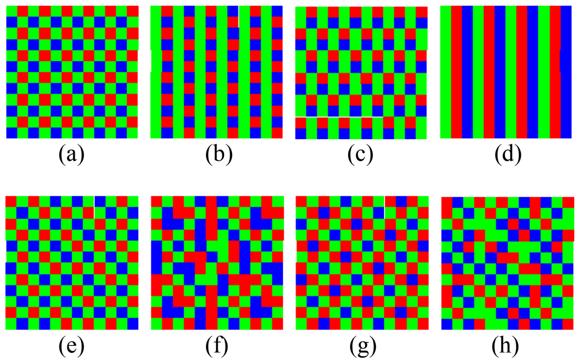


Fig. 1: RGB CFAs, (a) Bayer CFA (Jayachandran and Dhanasekaran, 2013), (b) Yamanaka CFA (Kakarala and Baharav, 2002), (c) vertical stripe HVS-based CFA (Lule *et al*., 2000), (d) vertical stripe CFA (Longère *et al*., 2002), (e) modified Bayer CFA (Longère *et al*., 2002); (f-g) pseudo-random CFA (Longère *et al*., 2002); (j) exiting HVS-based CFA (Longère *et al*., 2002)

Color Plane Interpolation algorithm (ACPI) proposed in Chung and Chan (2006). In this study, based on the framework of ACPI, a new heuristic demosaicing algorithm is proposed. This algorithm uses the variance of pixel color differences to determine the interpolation direction for interpolating the missing green samples. Simulation results showed more noises in the border of the image, so the border of the image is enhanced using some adaptive techniques (Kakarala and Baharav, 2002; Longère *et al*., 2002). Now Simulation results show that the proposed algorithm is superior to the latest demosaicing algorithms in terms of increased PSNR value.

METHODOLOGY

Proposed CFA pattern: The design and performance characteristics of the CFA are essentially determined by the type of a color system and the arrangements of the color filters in the CFA. The visual effect of an RGB color image is based on the weight given to the RGB components. The Proposed pattern is based on the actual weight of the Human visual system as shown in Fig. 2. In proposed Pattern we have given more weightage to the green samples, red samples weightage is same to the Bayer CFA pattern but small weighted to the blue samples. The visual effect of an RGB color image is based on the weight given to the RGB components. It is estimated that the sensitivity level of human eye (Longère *et al*., 2002; Alleysson *et al*., 2005) to read as 29.9%, green as 58.7% and blue as 11.4%. According to the well known Bayer's pattern, the red gives 25%, green is given 50% and blue gives 25%. Which is not close to the actual weights? We propose here a pattern which is close to the actual weight. It is proposed to give 25% to red, 62.5% for green and 12.5% for blue.
The main contributions of our proposed technique are:

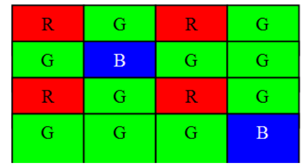


Fig. 2: Proposed CFA pattern

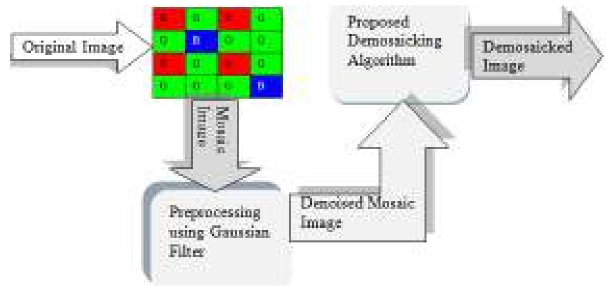


Fig. 3: Two stages denoising and demosaicking of the proposed method

- An efficient proposed human visual based color filter array applies to the original image and get demosaicked image
- Using Gaussian filter is applied to eliminate the noise and get the noise free image for further processing
- Using proposed iterative Enhanced directional based Demosaicking algorithm for Demosaicking process and get noise free Demosaicked image
- Used evaluation matrices PSNR, MSE to evaluate the performance of the proposed technique is compared with exiting Demosaicking algorithm

The following Fig. 3 describes the proposed Method performance step by step.

Pre processing: The mosaic image is subjected to a set of pre-processing steps so that the image gets transformed to be suitable for the further processing. The Pre-processing is used for loading the input color images to the MATLAB Environment and also it removes any kind of noise present in the input images. Here we make use of two step pre-processing procedures in which first the input image is passed through a Gaussian filter to reduce the noise and get a better image. Passing the image through the Gaussian filter also enhances the image quality.

Gaussian filter: A Gaussian filter (Haddad and Akansu, 1991; Jayachandran and Dhanasekeran, 2013) is a filter whose impulse response is a Gaussian function. Gaussian filters are developed to avoid overshoot of step function input while reducing the rise and fall time. This character is very much linked to the fact that the Gaussian filter has the minimum possible group delay. In mathematical terms, a Gaussian filter changes the input signal by convolution with a Gaussian function; this change is also called the Weierstrass transform. The Gaussian function is non-zero for $x \in [-\infty, \infty]$ and would supposedly need an infinite window length. The filter function is supposed to be the kernel of an integral transform. The Gaussian kernel is continuous and is not discrete. The cut-off frequency of the filter can be taken as the ratio between the sample rate F_s and the standard deviation σ:

$$F_c = \frac{F_s}{\sigma}$$

The 1D Gaussian filter is given by the equation:

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

The impulse response of the 1D Gaussian Filter is given by:

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

Here in the pre-processing step, the input image is passed through a Gaussian filter which results in reduction of the noise in the input image and also results in obtaining an image fit for further processing. Passing the image through the Gaussian filter also enhances the image quality.

Proposed EDGE oriented demosaicking algorithm: Based on the observations in Gunturk *et al*. (2002) and Hirakawa and Parks (2005), the proposed algorithm put its focus on how to effectively determine the interpolation direction for estimating a missing green component in edge regions and texture regions. In particular, variance of color differences is used in the proposed algorithm as a criterion to determine the interpolation direction for the green components. This proposed algorithm is shown in Fig. 4.

Interpolating missing green components: The color image has three color plane Red plane, Green plane and Blue Plane The missing green plane is calculate first, the remaining color plane are calculated Based on green plane i.e., the algorithm first calculate the two different planes $K_R \equiv G - R$ and $K_B \equiv G - B$. In each missing green samples is first calculated the weighted color difference value around the original pixel then to calculate the average of its neighboring color difference value. Here R, G, B is the original image pixel value, $\hat{R}$, $\hat{G}$, $\hat{B}$ are interpolated pixel value.

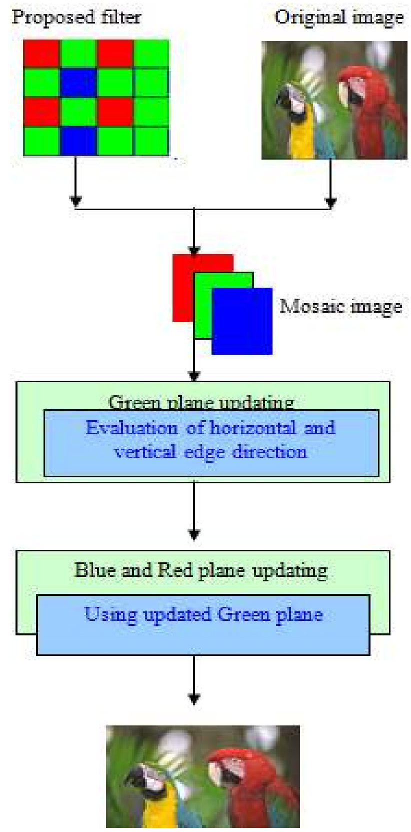


Fig. 4: Demosaicing algorithm

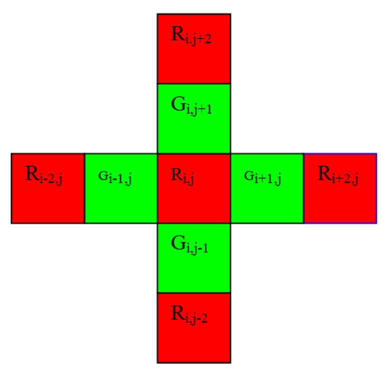
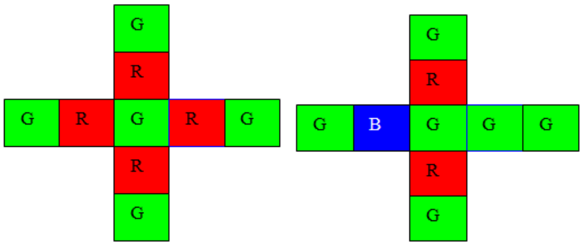


Fig. 5: The proposed CFA pattern having their centers at red CFA

As shown in Fig. 5, to obtain the $G_{i,j}$ value at the $R_{i,j}$ pixel, first to calculate the weights (α) along the four adjacent direction as follows:

$$\alpha_{i-1,j} = |R_{i-2,j} - R_{i,j}| + |G_{i-1,j} - G_{i+1,j}|$$
$$\alpha_{i+1,j} = |R_{i+2,j} - R_{i,j}| + |G_{i-1,j} - G_{i+1,j}|$$
$$\alpha_{i,j-1} = |R_{i,j-2} - R_{i,j}| + |G_{i,j-1} - G_{i,j+1}|$$
$$\alpha_{i,j+1} = |R_{i,j+2} - R_{i,j}| + |G_{i,j-1} - G_{i,j+1}|$$

Calculate The Four adjacent color difference is calculated as follows:



(a) (b)

Fig. 6: Proposed CFA pattern having their centers at green CFA samples

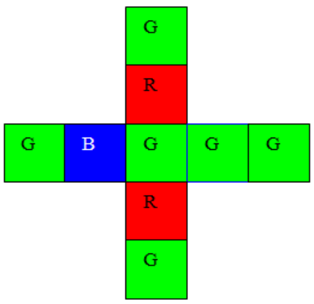


Fig. 7: Proposed CFA pattern having their centers at green CFA samples

$$Kr_{i-1,j} = G_{i-1,j} - 1/2\,(R_{i-2,j} + R_{i,j})$$
$$Kr_{i+1,j} = G_{i+1,j} - 1/2\,(R_{i+2,j} + R_{i,j})$$
$$Kr_{i,j-1} = G_{i,j-1} - 1/2\,(R_{i,j-2} + R_{i,j})$$
$$Kr_{i,j+1} = G_{i,j+1} - 1/2\,(R_{i,j+2} + R_{i,j})$$

where, r, g, b are missing components and R, G, B are original values. The weights are then assigned to the four adjacent color difference values Kx, as defined In Eq. (2), The estimating $Kx_{(i,j)}$ calculated as follows:

$$K1 = (Kr_{i-1,j}) / (1 + \alpha_{i-1,j})$$
$$K2 = (Kr_{i+1,j}) / (1 + \alpha_{i+1,j})$$
$$K3 = (Kr_{i,j-1}) / (1 + \alpha_{i,j-1})$$
$$K4 = (Kr_{i,j+1}) / (1 + \alpha_{i,j+1})$$

$$Kx_{(i,j)} = (K1 + K2 + K3 + K4) / ((1 + \alpha_{i-1,j}) + (1 + \alpha_{i+1,j}) + (1 + \alpha_{i,j-1}) + (1 + \alpha_{i,j+1}))$$

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

$$\hat{G}_{i,j} = R_{i,j} + Kx_{(i,j)}$$

Similarly The Green pixel value is calculated in Blue CFA components.

Interpolating missing red components at green CFA sample position: After interpolating all missing green components of the image, the missing red and blue components at green CFA sampling positions are estimated. Figure 6a and b shows the two possible cases where a green CFA sample is located at the center of a 5×5 block.

As for the case in Fig. 5a, the missing components of the red are calculated as Follows:

$$\hat{R}_{i,j} = G_{i,j} + [R_{i-1,j} - \hat{G}_{i-1,j} + R_{i+1,j} - \hat{G}_{i+1,j}]\,/2$$

As for the case in Fig. 5b, the missing components of the Red are calculated as Follows:

$$\hat{R}_{i,j} = G_{i,j} + [R_{i,j-1} - \hat{G}_{i,j-1} + R_{i,j+1} - \hat{G}_{i,j+1}]\,/2$$

Interpolating blue components at red and green CFA sample positions: Finally, the missing blue components at the red and green sampling positions are interpolated using the proposed algorithm. The blue components can be calculated based on the changing ratio of Green components in the same position. The missing blue sample, $b_{i,j}$, is interpolated by using the following algorithm:

$$B_{i,j} = G_{i,j} + \frac{B_{i-1,j} - G_{i-1,j} + B_{i+1,j} - G_{i+1,j}}{2}$$

As for the case in Fig. 7, the missing components of the red are calculated as follows:

Algorithm:

col = i mod 4
row = j mod 2
$B^{\wedge}_{i,j} = B_{(i-col+1,j-row+1)}$
$\Delta H_{i,j} = \hat{G}_{i,j} - \hat{G}_{(i-col+1,j-row+1)}$
NWeightage = $1.0*\Delta H_{i,j} / (\hat{G}_{(i-col+1,j-row+1)})$
$B^{\wedge}_{i,j} = B^{\wedge}_{i,j} + NWeightage * B^{\wedge}_{i,j}$

Interpolating missing blue (red) components at red (blue) sampling positions: Finally, the missing blue (red) components at the red (blue) sampling positions are interpolated. The missing blue sample, $b_{i,j}$ is interpolated by:

$$b_{i,j} = G_{i,j} + \frac{1}{4}\sum_{m=\pm 1}\sum_{n=\pm 1}(B_{i+m,j+n} - G_{i+m,j+n})$$

In a similar manner the missing red, $r_{i,j}$ is interpolated by:

$$r_{i,j} = G_{i,j} + \frac{1}{4}\sum_{m=\pm 1}\sum_{n=\pm 1}(R_{i+m,j+n} - G_{i+m,j+n})$$

SIMULATION RESULTS

Simulation was carried out to evaluate the performance of the proposed algorithm. The 12 digital

Fig. 8: Images used in the experiments (images are numbered from 1 to 12 in the order of left-to-right and top-to-bottom)

Table 1: PSNR performance comparison of different demosaicing methods

Image	Channel	POCS (Chung and Chan, 2006)	PCSD (Lule *et al*., 2000)	FSMU (Longère *et al*., 2002)	Proposal method
1	Red	28.4553	33.2122	35.2959	35.3456
	Green	32.9273	35.4010	39.1361	39.2313
	Blue	28.1420	33.0438	36.0125	35.9801
2	Red	31.4359	38.1016	34.4891	37.2280
	Green	33.3676	40.4894	40.9140	43.3307
	Blue	31.2941	37.7678	38.9587	37.1122
3	Red	29.2315	35.3110	36.3224	36.2321
	Green	33.2590	37.5920	39.3066	39.0213
	Blue	29.2253	34.6224	35.4082	34.9087
4	Red	30.2407	31.6644	30.6963	33.2211
	Green	33.5075	33.9435	34.3265	35.1234
	Blue	30.3382	30.9675	30.7919	29.2213
5	Red	27.2408	29.6016	33.3178	32.3112
	Green	32.3780	31.3597	35.5628	34.6781
	Blue	27.0262	29.3271	32.1294	31.3421
6	Red	31.1391	39.2980	38.5834	38.9786
	Green	33.7569	41.5848	41.7205	42.1237
	Blue	30.8272	41.5848	38.1094	37.6789
7	Red	29.7384	34.8317	35.7577	36.3456
	Green	31.6029	36.8330	38.9241	39.1034
	Blue	28.4014	34.1418	35.0944	35.3456
8	Red	26.4451	32.4751	33.7174	34.4567
	Green	28.0668	34.1149	36.1049	35.8902
	Blue	26.1760	30.8661	31.8199	31.6789
9	Red	30.7306	38.2405	37.5531	38.2345
	Green	32.4580	40.0180	40.2358	40.2456
	Blue	30.5046	37.8739	37.3851	37.1234
10	Red	28.8603	33.3901	33.5461	33.1235
	Green	32.1476	35.0823	36.3242	36.4567
	Blue	30.5046	33.0352	33.6770	32.7567
11	Red	27.7125	35.3082	35.9037	35.6789
	Green	30.7234	38.8958	39.2317	40.1235
	Blue	28.4945	36.3010	36.3023	36.5674
12	Red	27.2650	38.2058	37.5343	38.4567
	Green	28.8643	39.8856	40.1988	41.5678
	Blue	28.4662	36.4789	35.2987	35.0987

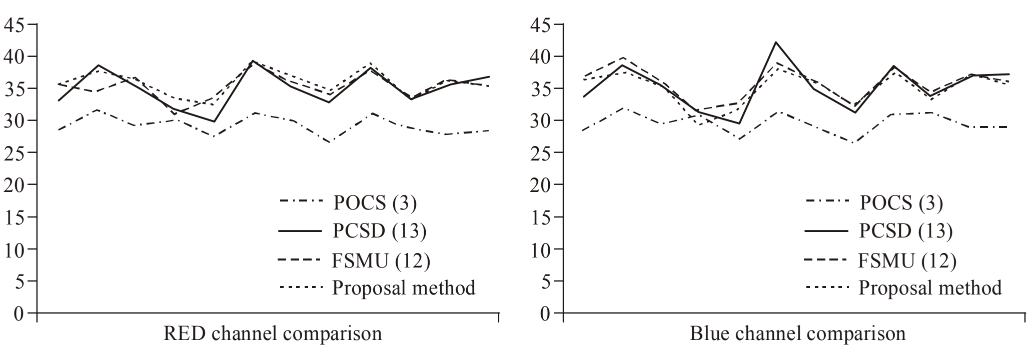


Fig. 9: PSNR value comparisons of red and blue channel to exiting demosaicking algorithm

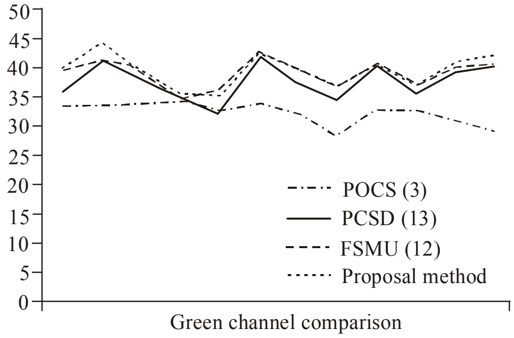


Fig. 10: PSNR value comparisons of green channel to exiting demosaicking algorithm

color images shown in Fig. 8 were used to generate a set of testing images. The Peak Signal-to-Noise Ratio (PSNR) was used as a measure to quantify the performance of the demosaicing methods. In particular PSNR, it is defined as:

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

where,

$$CMSE = (1/3HW)\sum_{i=r,g,b}\sum_{y=1}^{H}\sum_{x=1}^{W}((I_0(x,y,i) - I_r(x,y,i))^2$$

I_o and I_r represent, respectively, the original and reconstructed images of size H×W each.

Table 1 shows the performance of various algorithms. The proposed algorithm provided the best performance among the evaluated algorithm. The proposed algorithm is developed based on the fact that the interpolation direction for each missing green sample is critical to the final demosaicing result. The following Fig. 9 and 10 shows the experimental results of exiting and our proposed method.

CONCLUSION

In this study, a new color filter array based on HVS and iterative edge based demosaicking algorithm with enhanced border and edge was presented. It makes use of the color difference variance of the pixels located along the horizontal axis and that along the vertical axis in a local region to estimate the interpolation direction for interpolating the missing green samples. The high-performance arises from the introduction of a weighted edge interpolation and the well designed stopping strategy. With them, the proposed algorithm has a good initial condition and can terminate iteration early. Simulation results show that the proposed algorithm is able to produce a subjectively and objectively better demosaicing results as compared with a number of advanced algorithms.

REFERENCES

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

Bayer, B.E., 1976. Color Imaging Array. U.S. Patent 3 971 065, July 1976.

Chung, K.H. and Y.H. Chan, 2006. Color demosaicing using variance of color differences. IEEE T. Image Process., 15(10): 2944-2955.

Dillon, P.L.P., D.M. Lewis and F.G. Kaspar, 1978. Color imaging system using a single CCD area array. IEEE J. Solid-St. Circ., 13(1): 28-33.

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

Gunturk, B.K., J. Glotzbach, Y. Altunbasak, R.W. Schaffer and R.M. Murserau, 2005. Demosaicking: color filter array interpolation. IEEE Signal Proc. Mag., 22(1): 44-54.

Haddad, R.A. and A.N. Akansu, 1991. A class of fast gaussian binomial filters for speech and image processing. IEEE T. Acoust. Speech., 39: 723-727.

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

Jayachandran, A. and R. Dhanasekeran, 2012. Efficient demosaicking of color images using theoretic reconstruction technique. Proceeding of the IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), pp: 282-286.

Jayachandran, A. and R. Dhanasekaran, 2013. Automatic detection of brain tumor in magnetic resonance images using multi-texton histogram and support vector machine. Int. J. Imag. Syst. Tech., 23: 97-103.

Kakarala, R. and Z. Baharav, 2002. Adaptive demosaicing with the principal vector method. IEEE T. Consum. Electr., 48(11): 932-937.

Longère, P., X. Zhang, P.B. Delahunt and D.H. Brainard, 2002. Perceptual assessment of demosaicing algorithm performance. Proc. IEEE, 90(1): 123-132.

Lukac, R. and K.N. Plataniotis, 2005a. Data-adaptive filters for demosaicking: A framework. IEEE T. Consum. Electr., 51(2): 560-570.

Lukac, R. and K.N. Plataniotis, 2005b. Fast video demosaicking solution for mobile phone imaging applications. IEEE T. Consum. Electr., 51(2): 675-681.

Lule, T., S. Benthien, H. Keller, F. Mutze, P. Rieve *et al*., 2000. Sensitivity of CMOS based imagers and scaling perspectives. IEEE T. Electron. Dev., 47(11): 2110-2122.

Parmar, M. and S.J. Reeves, 2004. A perceptually based design methodology for color filter arrays. Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'04), 3: 473-476.

Parulski, K. and K.E. Spaulding, 2002. Color Image Processing for Digital Cameras. In: Sharma, G. (Ed.), Digital Color Imaging Handbook. CRC Press, Boca Raton, FL., pp: 728-757.

Wu, X. and N. Zhang, 2004. Primary-consistant soft-decision color demosaicking for digital cameras. IEEE T. Image Process., 13(9): 1263-1274.