

A Self-Learning Approach to Edge-Adaptive Demosaicking Algorithm for CFA Images

Shiny.A1,Nagarajan.G2

1Shiny.A. Author is currently pursuing M.Tech(Information Technology) in Vins Christian College of Engineering. e-mail:shinyarsha@gmail.com,

2Nagarajan.G. Author is currently working as Assistant professor, Department of IT in VINS Christian College of Engineering.

Abstract:

Most of the current digital cameras will capture only one color at each of the pixel using a single sensor overlaid with a color filter array (CFA). In order to recover a full color image from a mosaic pattern, specific methods will be used to restore the two missing color values at each of the pixels in the Bayer pattern. The restoration process to find out the missing color samples is known as color demosaicking. Color-difference based edge-adaptive filtering will reduce the error in the red and blue channel by exploiting the high frequency information of the green channel in the proposed enhanced edge-adaptive demosaicking algorithm. By using the self-learning approach via support vector regression, it extracts the image dependent information in constructing the learning model. It does not require any additional training data externally. Hence the discussed method will show that the proposed system will provide state of the art in both subjective and objective image quality.

Keywords—Digital camera, Color demosaicking, color filter array, adaptive filtering, color artifacts, self-learning, support vector regression.

I. INTRODUCTION

With the rapid progress of technology, a digital still camera uses a charge-coupled device (CCD) or a complementary metal oxide semiconductor (CMOS) as its sensor. Because the sensor can detect only the intensity of light and is not responsive to the differences of colors, so a color filter must be installed in front of the sensor when capturing images. Generally, color filters work on the principle of color separation of the three primary colors (R, G, B) and are placed in front of three sensors, so that the RGB values captured respectively by the three sensors can be combined to form a full-color image. The major limitation of using three sensors will increase the cost, power consumption, and size. To overcome from these limitations, most digital cameras capture one primary color at each pixel by a single sensor overlaid with color filter array. The most popular CFA pattern is Bayer pattern in fig 1. As a result, each pixel only acquires the intensity (brightness) of one of the three-color elements R, G and B while losing the other two color elements. Therefore, an interpolation must be performed on the result obtained through the CFA, so as to reconstruct the missing color elements in each pixel. Thus the process of reconstructing the missing color elements is called demosaicking or CFA interpolation.

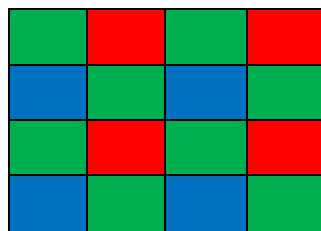


Fig 1: The Bayer pattern of color mosaic for digital cameras.

In literature, various approaches for demosaicking algorithms have been introduced. The initial work is based on Simple interpolation techniques, such as nearest-neighbour, bilinear, and bicubic interpolation. These single-channel algorithms usually introduce severe color artifacts and blurring around the sharp edges. These drawbacks indicate the need for more specialized algorithm for better performance. Hence better performance was achieved by edge-directed interpolation approach [1,2,3]. This approach will perform the interpolation along the image edges and produce high Quality visual results, especially in reconstructing the sharp edges of the demosaicked images. However, in regions of fine details and edges tend to be short and in different direction, these algorithm will introduce an undesirable errors and generate color artifacts. Existing algorithm are unable to resolve the

color artifacts in high-frequency region such as edges or fine textures to obtain demosaicked results with high visual quality.

Due to the richness of natural images, direct analytical analysis and modeling of image processing models are very challenging. Recently, researchers also suggested tackling the image demosaicking problem using machine learning approaches. For instance, Zhang et al. [4] first calculate the intra and inter-band interpolation independently, and utilize linear minimum mean-square error (LMMSE) and support vector regression (SVR) to determine the weighting factors for the intra and inter-band of the final fused image. Wu et al. [5] proposed a sparsity-based method. They introduce the sparse representations for both intra and inter-channel, and integrate them into an ℓ_1 minimization approach to solve the inverse program. The common drawback of most of the prior learning-based approaches is the need to collect additional training data. To remedy these above problems, we propose an edge-adaptive demosaicking algorithm (EAD) and self-learning color demosaicking method in this paper. The edge-adaptive demosaicking algorithm (EAD) consists of color-difference based edge-adaptive low pass filtering to reproduce the color values by exploiting the green channel information for making high frequency components of red and blue channels similar to the green channels. The red and blue channels are first reconstructed using bilinear interpolation and then the green channel is reconstructed. Once the three color plane is reconstructed the next step is to introduce the color-difference based edge-adaptive filtering. Another advantage existing image interpolation method can be combined with the proposed algorithm to reconstruct the green channel. It doesn't ensure that the demosaicked output thus obtained is appropriate. Hence self-learning approach is introduced with the proposed algorithm. In this it doesn't need any additional training data like learning approaches. It will provide an approximation of calculations and machine itself extract their needed information. Given a mosaicked input image, its full color version is first estimated and we downsample this interpolated image into smaller scales as the ground truth. A color filter array is then applied on each coarser scale image of the image pyramid to obtain its spatial sampled mosaicked image. The relationship between the image pyramid and the corresponding mosaicked images is learned through SVR. Finally, the full color image of the test mosaicked image could be obtained by employing the learned SVR function on the test mosaicked image. We compare the proposed demosaicking algorithm results with the self-learning outcome

by SVR prediction. Experimental results show that our proposed method outperforms these state-of-the-art techniques in both subjective and objective image quality measures.

The rest of the paper is organized as follows. Section 2 presents the proposed model. It describes both the edge-adaptive demosaicking algorithm and the Self-learning demosaicking model. Section 3 presents our proposed edge-adaptive demosaicking algorithm with our self-learning demosaicking algorithm and also it describes the working concepts of the proposed algorithm. Experimental results are reported in Section 4. Finally, conclusions and references are given in Section 5 and section 6.

II. PROPOSED MODEL

Our proposed system describes two concepts. First describes about edge adaptive demosaicking algorithm and second describes the self-learning demosaicking. In section 2.1, we discuss about, how we reduce the color artifacts by using proposed demosaicking algorithm. Section 2.2 presents, how we construct a self-learning model for image demosaicking without the need to collect any training image data in advance and also how the learning and prediction can be achieved via Support vector regression in color demosaicking with the demosaicked results.

A. Edge-adaptive demosaicking algorithm

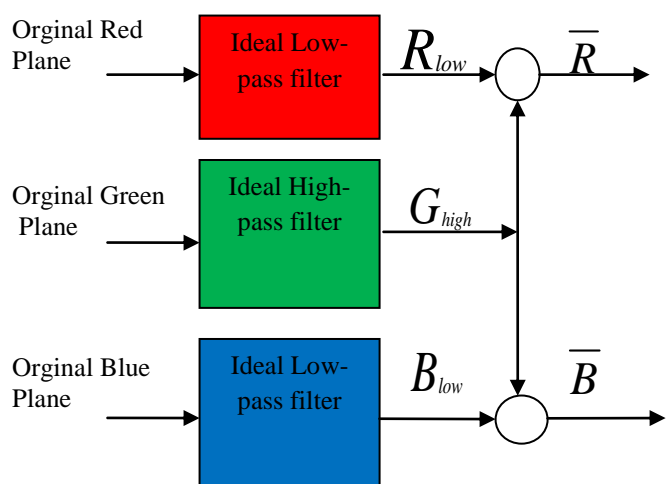


Fig 2. Flowchart for demonstrating the color-difference model.

The main aim of (EADA) edge-adaptive demosaicking algorithm is to effectively reduce the color artifacts in demosaicked images from a CFA. It also reduces the error in the red and blue channel by exploiting the high frequency information

of the green channel. To achieve this, color difference based edge-adaptive filtering is designed to reproduce the color values by exploiting the green information.

i. Color-difference model to demosaicking

In a Bayer pattern, green samples are obtained on a quincunx, while the red and blue samples are obtained in rectangular lattices which is shown in fig 1. The density of red and blue samples is one half of the green samples and also aliasing error of high frequency components in green channel is likely to be less than in red and blue channels. Hence the Bayer pattern is most popular method used in the demosaicking concept. In Bayer pattern green color will be twice than the other colors. The main problem in demosaicking is color artifacts in high frequency regions are caused by aliasing in the red and blue channels. Fig.2 shows the flowchart for demonstrating the assumption of color-difference model. In this it replace the high-frequency components of red and blue channels by using those green planes and then compare mean squared error (MSE) between the original and reconstructed color planes. A low-pass filter is used for red and blue planes and a high-pass filter is used for green planes. After filtering in each color plane, the red and blue channels are reconstructed by adding the high-frequency components of the green channel to their low-frequency components. The MSE is reduced effectively by adding the high-frequency region of the green plane to the low pass filtered red or blue planes. This denotes that the high-frequency regions of red and blue planes are similar to the high frequency region of the green plane.

ii. Edge-adaptive low-pass filtering

Let $[R_i^d \ G_i^d \ B_i^d]$ denotes the three color planes of the initial demosaicked image. The color-difference planes are given by

$$\begin{aligned} R_g &= R_i^d - G_i^d \\ B_g &= B_i^d - G_i^d \end{aligned} \quad (1)$$

Where $[R_i^d \ and \ B_i^d]$ are the initial estimated red and blue channels obtained from bilinear interpolation and G_i^d is the green channel obtained by any interpolation method. The filtering involves two steps. First, edge adaptive low-pass filtering of the red (blue) values over the original blue (red) pixels as shown in Fig 3(a). Second, edge adaptive low-pass filtering of the red (blue) values over the original green pixels as shown in Fig 3(b). Here the same

concepts is used for both R_g and B_g color -difference planes, only the concepts of R_g will be described in below.

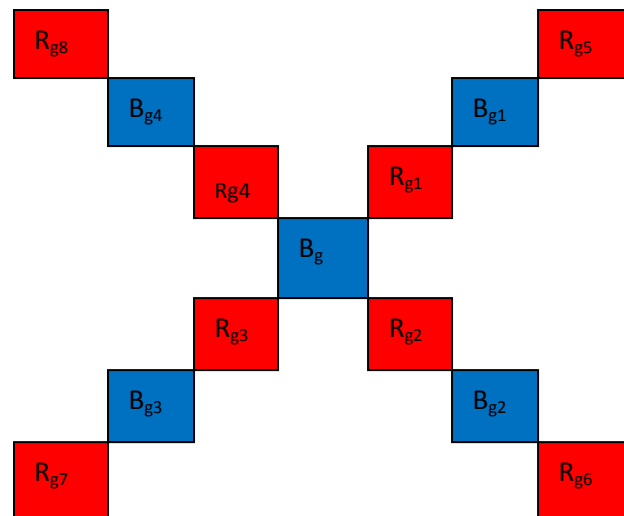


Fig.3 (a) The red value on blue pixel

By referring the Fig 3(a) and 3(b), the color-difference value of R_g red pixel at the center position is filtered by,

$$R_g = \frac{e_{a1}R_{g1}^{\wedge} + e_{a2}R_{g2}^{\wedge} + e_{a3}R_{g3}^{\wedge} + e_{a4}R_{g4}^{\wedge}}{e_{a1} + e_{a2} + e_{a3} + e_{a4}} \quad (3)$$

Where $R_{g1}^{\wedge} \sim R_{g4}^{\wedge}$ are the color-difference adjusted values and $e_{a1} \sim e_{a4}$ are the edge indicators corresponding to each color-difference adjusted value. Edge indicators are used to find the type of edges. These edge indicators are defined as a decreasing function of the directional derivative of the center point and its neighbouring points. In the case of Fig.3(a), the edge indicators of e_{a1} for the red value on blue pixel can be found out by the sum of the difference of $\frac{R_{g1} - R_{g3}}{2\sqrt{2}}$ and difference of $\frac{R_{g5} - 2R_{g1} - R_{g3}}{2\sqrt{2}}$.), the edge indicators of e_{a2} for the red value on blue pixel can be found out by the sum of the difference of $\left| \frac{R_{g4} - R_{g2}}{2\sqrt{2}} \right|$ and difference of $\frac{R_{g4} - 2R_{g2} - R_{g6}}{2\sqrt{2}}$. Similarly e_{a3} and e_{a4} can be found out for the red value on blue pixel. Then the edge indicators $e_{a1}, e_{a2}, e_{a3}, e_{a4}$ are defined by,

$$e_{a1} = \frac{1}{1 + \left| \frac{R_{g1} - R_{g3}}{2\sqrt{2}} \right| + \left| \frac{R_{g5} - 2R_{g1} - R_{g3}}{2\sqrt{2}} \right|} \quad (4)$$

$$e_{a2} = \frac{1}{1 + \left| \frac{R_{g4} - R_{g2}}{2\sqrt{2}} \right| + \left| \frac{R_{g4} - 2R_{g2} - R_{g6}}{2\sqrt{2}} \right|} \quad (5)$$

$$e_{a3} = \frac{1}{1 + \left| \frac{R_{g1} - R_{g3}}{2\sqrt{2}} \right| + \left| \frac{R_{g1} - 2R_{g3} - R_{g7}}{2\sqrt{2}} \right|} \quad (6)$$

$$e_{a4} = \frac{1}{1 + \left| \frac{R_{g4} - R_{g2}}{2\sqrt{2}} \right| + \left| \frac{R_{g8} - 2R_{g4} - R_{g3}}{2\sqrt{2}} \right|} \quad (7)$$

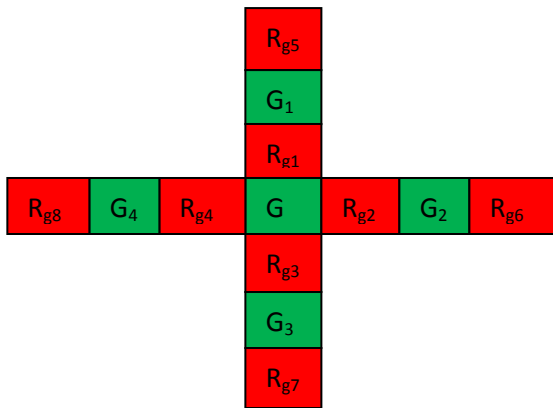


Fig 3(b). The red value on green pixel

In the case of Fig.3(b), the edge indicators are given by,

$$e_{a1} = \frac{1}{1 + \left| \frac{R_{g3} - R_{g1}}{2} \right| + \left| \frac{R_{g3} - 2R_{g1} - R_{g5}}{2} \right|} \quad (8)$$

$$e_{a2} = \frac{1}{1 + \left| \frac{R_{g2} - R_{g4}}{2} \right| + \left| \frac{R_{g6} - 2R_{g2} - R_{g4}}{2} \right|} \quad (9)$$

$$e_{a3} = \frac{1}{1 + \left| \frac{R_{g3} - R_{g1}}{2} \right| + \left| \frac{R_{g7} - 2R_{g3} - R_{g1}}{2} \right|} \quad (10)$$

$$e_{a4} = \frac{1}{1 + \left| \frac{R_{g2} - R_{g4}}{2} \right| + \left| \frac{R_{g2} - 2R_{g4} - R_{g8}}{2} \right|} \quad (11)$$

The color-difference adjusted values $R_{g1} \sim R_{g4}$ are derived based on the assumption that the difference of neighbouring color-difference values along an interpolation direction is constant and edge indicators e_{a1} to e_{a4} denotes the edges.

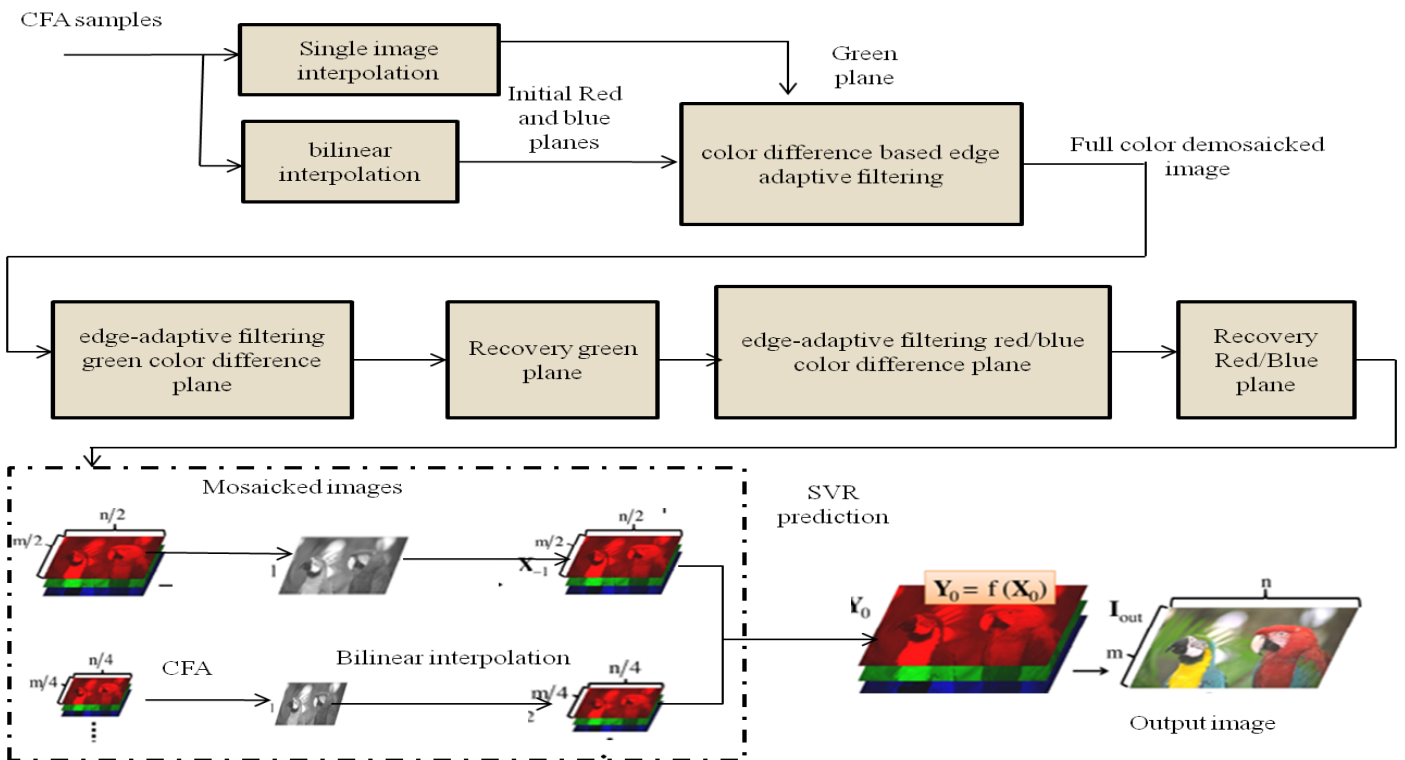


Fig 4 complete steps for proposed architecture

iii. Green channel adaptive interpolation

This approach is used to reconstruct the green channel from the CFA samples. In Bayer pattern the green plane has the most spatial information and it has great influence on the quality of the image. In order to reconstruct the green plane with quality, we propose a non-linear procedure for choosing the direction of interpolation. The green value of center pixels are to be estimated from its four surrounding green pixel. In smooth region, central missing green value is then estimated by weighted sum. In the edge region, the central missing green value is then carried out by selecting weighted sum in horizontal and vertical directions.

B. Self-learning demosaicking

In learning-based models, there is a need to collect the training data. But in this approach, it doesn't need to collect any additional training data in advance. Machine itself extracts the image dependent information. Similarly no assumption such as image priors needs to be considered in this method. In section 2.2.1 it describes, how we construct a self-learning model for image demosaicking without the need to collect any training image data in advance. In Section 2.2.2 it describes about the learning and prediction with support vector regression (SVR) in our color demosaicking framework

i. Construction of training set

In learning models such as Super-Resolution, super-denoising etc there is a need to collect additional training data. In this method, we downgrade the input image into different smaller scales as ground truth data, as inspired by [6, 7, 8] for self-learning purpose. With such a proposed framework, we are able to construct the relationships between different models of CFA pattern and the corresponding color information based on the input mosaicked image and thus the observed models can be applied for color demosaicking. We now discuss how we utilize the input image to collect training data and ground truth information for self-learning purposes. Take an input mosaicked image of size $m \times n$ pixels, where each pixel only contains one of the R, G, or B values. To obtain an initial demosaicked image, we apply an interpolation algorithm to an input mosaicked image. However, since interpolation-based approaches might not achieve satisfactory demosaicking results due to blurring, aliasing etc. Hence we do not apply the

above demosaicked results directly for synthesizing the final output. Instead, we downgrade its resolution into a smaller scale of size $\frac{m}{2} \times \frac{n}{2} \times 3$. In the downsampled image apply the CFA and then the bilinear interpolation. Once it is completed, the next step is to compare the interpolated image with its downsample version and it can be considered as the ground truth demosaicked image of size $\frac{m}{2} \times \frac{n}{2}$ pixels.

ii. Learning of demosaicking models via SVR

Given a mosaicked input image, we are able to collect input data X_i and their ground truth Y_i . To design a self-learning algorithm for demosaicking purposes, we need to model the relationships between the image patches X_i and the associated pixels Y_i . Let us take a mosaicked image as input in fig 4. As illustrated in Fig. 5, there exist four different types of Bayer patterns in this image, depending on the locations of the R, G, and B values in the image patch of interest. As a result, we have four different types of input data available in the input mosaicked image X_i . Since our goal is to learn/predict the missing pixel labels in the center of each Bayer pattern input and we need to formulate two distinct learning models to recover the two missing pixel values, and thus we will have a total of 4×2 learning models $Y = f(X)$ to be observed in our proposed framework. In this, machine itself extracts the image dependent information and no additional training data is needed. It also provides the approximation of calculation also provides the approximation of calculation. For the j -th patch in input image of size $h \times h$ Pixels, the corresponding training input instance in X_i is defined as,

$$x_j^t = [R_j^x(1,1), R_j^x(1,2), \dots, G_j^x(1,1), \dots, B_j^x(h,h)] \dots (12)$$

Where $t \in 1,2,3,4$ indicates the four types of Bayer pattern block. $M \in \{R, G, B\}$ indicates the two missing Red, Green, or Blue values at the patch center. The above equation demonstrates the training input instance for the j -th patch in input image of size $h \times h$ Pixels. Support vector regression (SVR) is an effective regression technique due to its generalization ability. Once the above training process is complete, we apply the learned SVR models for refining the image X_0 into Y_0 , as shown in Fig. 4. First, we examine its Bayer pattern block and select the associated SVR model to predict the center missing pixel values. Once this prediction is finished, we successfully refine the image X_0 into Y_0 , and thus

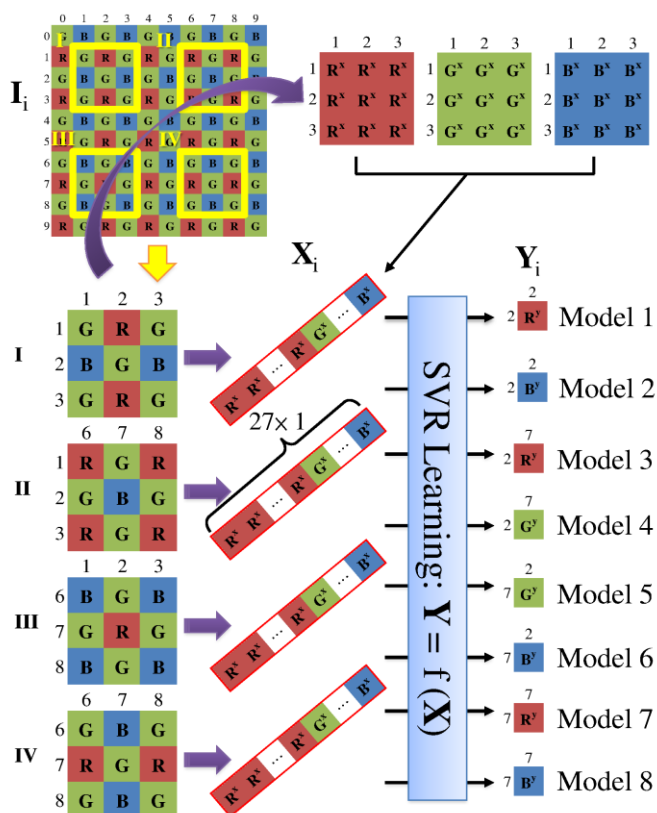


Fig.5 learning of eight demosaicking model via SVR

the demosaicked output is obtained. By [3] is applied in our framework for producing the final output I_{out} . Thus the full color version color demosaicked image is obtained as output.

III. EDGE-ADAPTIVE DEMOSAICKING BY SELF- LEARNING MODEL

In this section, the proposed model in fig 4 presents, how to reduce the color artifacts and how it provide a better demosaicked results. In order to achieve this self-learning model is introduced. In this model, machine itself will extract their needed information from the image priors and no training data is needed in advance. By this approach it will learn or predict the outcome of learned model with the demosaicked results by using SVR prediction to provide a better demosaicked image results. Edge-adaptive demosaicking algorithm will reduce the color artifacts by exploiting the high-frequency components of the green channel. These color artifacts are caused by aliasing error in the high-frequency region such as edges or fine textures. In order to reduce this color artifacts, color-difference based edge-adaptive filtering is designed to reproduce the color. The main advantage is that existing image interpolation can be combined with the proposed model to reconstruct the green channel. Thus the full color demosaicked image $R_i^d G_i^d B_i^d$ is obtained. The key concept of using this color-difference based filtering is to replace the high-frequency components of red and blue channels by using those green planes and then red

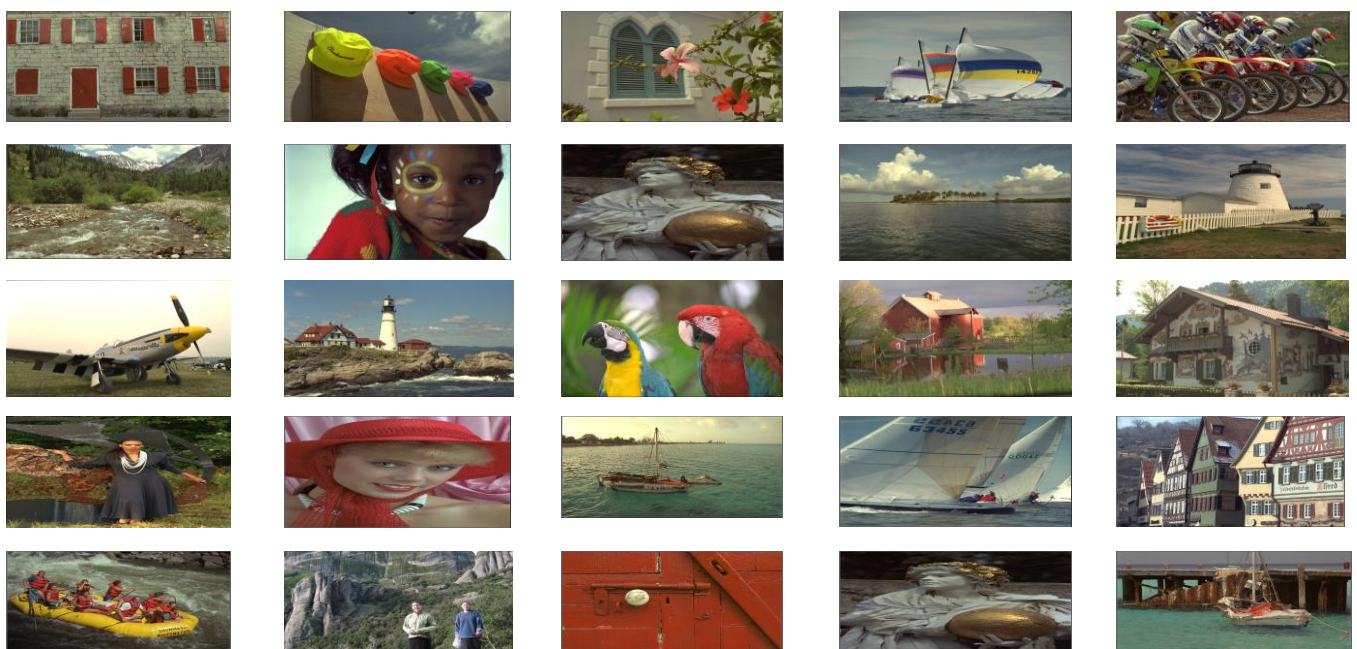


Fig .6 Test images of Kodak dataset used in the experiment.

red and blue channels by using those green planes and then compare mean squared error(MSE) between the original and reconstructed color planes. A low-pass filter is used for red and blue planes and a high-pass filter is used for green planes. After filtering in each color plane, the red and blue channels are reconstructed by adding the high-frequency components of the green channel to their low-frequency components. The MSE is reduced effectively by adding the high-frequency region of the green plane to the low pass filtered red or blue planes. This denotes that the high-frequency region of red and blue planes are similar to the high similar to the high frequency region of the green plane. Therefore, demosaicked image is obtained by the edge-adaptive demosaicking algorithm and demosaicked image thus obtained can be input for the self-learning demosaicking model.

In self-learning demosaicking model, the demosaicked input image is considered as the ground truth image of size $\frac{m}{2} \times \frac{n}{2}$ pixels and it is downsample into number of smaller image as $\frac{m}{2} \times \frac{n}{2} \times 3$. Once the image is downsample, CFA is applied and then performs the bilinear interpolation for the downsample mosaicked image. Once it is completed, the next step is to compare the interpolated demosaicked image with its downsample version by SVR prediction. SVR prediction will compare both the results and predict the approximation of calculation. Main goal is to learn/predict the missing pixel labels in the center of each Bayer pattern. Once this prediction is finished, we successfully refine the image X_0 into Y_0 , and thus the demosaicked output is obtained with high quality.

IV. EXPERIMENTAL RESULTS

This section compares the performance of the proposed and other methods. To evaluate the performance of our proposed methods, we compare their demosaicked results with those of five existing Demosaicking methods- Hamilton and Adams, Enhanced effective color interpolation, Dubois, Edge-adaptive demosaicking and Gradient. It shows some resultant image to observe strong and weak points of each algorithm. The proposed algorithm is tested on the Kodak image set. The test set consists of 24 images with 512-by-768 pixel resolution. The images are first downsample in Bayer pattern and then interpolated back to three channels using the proposed algorithm. Once the edge adaptive interpolation is done on the input image, the full color demosaicked image is obtained and the output of the demosaicked result will provide an input for the self-learning demosaicking model.

The demosaicked input image is downsample into number of smaller image as $\frac{m}{2} \times \frac{n}{2} \times 3$. Once the image is

downsample, CFA is applied and then performs the bilinear interpolation for the downsample mosaicked image. Once it is completed, the next step is to compare the interpolated demosaicked image with its downsampled version by SVR prediction. SVR prediction will compare both the results and predict the approximation of calculation. Hence the output obtained from the SVR prediction be a demosaicked image. Then this self-learned interpolated image is compared to the original images and results are reported in terms of PSNR measure. Results of the proposed algorithm are compared with Hamilton and Adams (HA) [9], enhanced effective color interpolation (EECI) [10], frequency domain demosaicking by Dubois (Dubois) in [15]. Adaptive filtering method (AF) [2] and edge-adaptive demosaicking (EAD) in [4]. Thus the experimental results show that the quality and performance of the image is improved.

TABLE 1
PSNR comparison among interpolation method

Image no	HA	EECI	Dubois	EAD	Gradient	proposed
1	34.69	40.65	41.02	41.32	41.97	43.09
2	40.86	43.99	43.13	44.92	44.43	45.12
3	42.18	45.63	44.10	45.12	44.81	45.89
4	40.63	43.46	43.42	44.72	43.13	44.54
5	35.96	40.96	40.52	40.31	40.72	42.45
6	36.08	41.18	43.15	42.74	43.11	44.56
7	42.29	45.46	44.91	44.80	44.62	46.26
8	33.46	38.65	38.60	39.52	39.19	41.34
9	41.47	45.49	44.75	45.13	45.72	46.05
10	41.36	45.59	45.16	45.34	45.60	47.23
11	37.17	42.18	42.48	42.82	43.08	44.89
12	42.31	45.76	46.02	46.42	46.18	47.65
13	30.65	36.86	37.52	37.05	37.99	41.34
14	37.27	40.56	38.78	40.12	39.70	40.32
15	39.44	42.84	42.86	43.22	42.50	44.67
16	39.67	44.90	46.67	45.63	46.07	48.78
17	39.41	44.08	44.05	43.56	44.41	45.34
18	34.96	39.84	39.85	39.53	40.12	43.23
19	38.38	42.87	43.23	43.46	43.78	44.65
20	39.78	44.24	43.45	44.07	44.35	46.54
21	36.37	41.90	41.88	42.04	42.70	43.06
22	37.87	40.94	40.81	40.71	40.99	42.13
23	43.40	45.61	44.78	45.53	45.41	46.78
24	33.52	37.64	38.33	37.71	37.91	39.35

A. PSNR comparison

The PSNR values of these methods are summarized in Table 1. The table shows that the proposed algorithm outperforms others in terms of PSNR results with all test images. Thus, the proposed algorithm can get higher PSNR than other methods. Hence the performance of the proposed method is improved than the existing demosaicking algorithms.

V. CONCLUSION

This paper presents an edge-adaptive CFA demosaicking algorithm along with self-learning demosaicking model. The proposed edge-adaptive demosaicking algorithm not only reduce the color artifacts in edge regions and fine textures but also it reduce the aliasing, blurring and provides a better demosaicked results with quality. But we can't ensure that the demosaicked image thus obtained should be appropriate. In order to achieve this, a self-learning based demosaicking concept is introduced. In this method, machine itself extracts the image dependent information and no training data is needed in advance. Another advantage is that, no assumption such as image priors need to be considered and provides an approximation of calculation. Unlike other methods, in this approach prediction can be done by SVR. Thus it provide a better quality image with lower complexity and also it improves the processing speed and PSNR value. Thus, Quantitative and qualitative empirical results confirmed that our approach resulted in promising demosaicked images while outperforming state-of-the-art demosaicking techniques.

REFERENCES

- [1] Alain Hoer and Djemel Ziou, "An Edge-Sensing Generic Demosaicking Algorithm with Application to Image Resampling" in *IEEE Trans*, 2011.
- [2] chi-yi Tsai, Kai-Tai Song, "A new edge-adaptive demosaicking algorithm for color filter arrays" in *IEEE Trans*, 2006
- [3] Fang-Lie He, Yu-Chiang Frank Wang, and Kai-Lung Hue, "A self-Learning Approach To Color Demosaicking via Support Vector Regression" in *IEEE Trans*, 2013.
- [4] chi-yi Tsai, Kai-Tai Song, "A new edge-adaptive demosaicking algorithm for color filter arrays" in 2007 Elsevier.
- [5] Wen-Jan Chen *, Pei-Yu Chang, "Effective demosaicking algorithm based on edge property for color filter arrays", in 2011 Elsevier
- [6] Y. Nomura and S. K. Nayar, "A vq-based demosaicking by self-similarity," in *IEEE ICIP*, 2007
- [7] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *IEEE ICCV*, 2009.
- [8] C Y. Yang, J.-B. Huang and M.-H. Yang, "Exploiting self-similarities for single frame super resolution" in *ACCV*, 2010.
- [9] J.F Hamilton, and J.E Adams, "Adaptive color plane interpolation in single sensor color electronic camera" U.S Patent 5,629,734, 1997.
- [10] L. Chang and Y.P Tan "Effective uses of spatial and spectral correlation for color filter array demosaicking", *IEEE signal processing*, 2004.
- [11] X. Wu, D. Gao, G. Shi, and D. Liu, "Color demosaicking with sparse representations," in *IEEE ICIP*, 2010.
- [12] H. Ducker, C. J. C. Burges, L. Kaufman, A. J. Smola, and Vapnik, "Support vector regression machines," in *Advances in NIPS*, 1996.
- [13] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM TIST*, 2011.
- [14] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM TIST*, 2011.
- [15] E Dubios, "Frequency-domain methods for demosaicking of Bayer sampled color images", *IEEE signal processing*, 2005.