Hybrid Directional Weight-based Demosaicking for Bayer Color Filter Array

Yonghoon Kim and Jechang Jeong

Abstract— This paper proposes and improves gradient estimation techniques in order to suppress the common demosaicking artifacts and introduces hybrid directional weights for adaptive directional interpolation. The hybrid weights are calculated using color correlation and intensity differences, and these weights apply to the whole interpolation process. Experimental results demonstrate that the proposed algorithm outperforms recent demosaicking methods on 42 images of Kodak and McM datasets.

Keywords— Bayer pattern CFA, CFA interpolation, demosaicking, directional interpolation.

I. Introduction

CONVENTIONAL digital cameras, which use single chip, exploit color filter array (CFA) to capture different spectral components. CFA interpolation is the process that reconstructs the full color image from the CFA captured image. CFA interpolation is a key algorithm for low-cost digital cameras which uses a single-chip. These cameras take images only one color per a pixel position although three color values are required for a full color image. Therefore to record the color channel effectively, single-chip digital cameras use a CFA. The resulting image is given as a gray-scale mosaic-like image. The CFA interpolation, which is called demosaicking, is the process that reconstructs a three color layer form CFA pattern image. The most commonly exploited pattern is Bayer pattern [1], which is shown in Fig. 1. The reason why green (G) values have a higher sampling rate than red (R) and blue (B) color channels is that the human visual system is sensitive to medium wavelength ranges which match with green color spectrum.

There are two main ideas which are exploited in a modern CFA interpolation algorithm. The first idea is interpolating the green channel first because it has twice as many numbers as other color channels. Therefore, a green channel suffers less from aliasing and an accurately interpolated green channel helps reconstruction of other color channels. The second idea is using the correlation of green-red and green-blue. Based on this concept, several demosaicking solutions have been proposed. In [2], the spectral correlation is modeled as constant color ratio rule and other concept, which is constant color difference rule, is proposed [3-4].

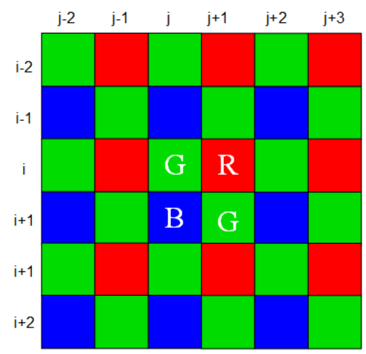


Fig. 1. Bayer color filter array pattern.

In [5], inter-channel correlation is used for color interpolation and one-step iteration algorithm is exploited to reduce the complexity. Chung *et al*. [6] tried to preserve details in texture region by using hard direction decision based on color difference variance. In [7], Paliy *et al*. modified the algorithm of [6] by proposing scale adaptive filtering based on local polynomial approximation (LPA). In [8], authors adopted the concept of directional demosaicking noise and proposed directional linear minimum mean square-error estimation (DLMMSE) algorithm.

Several methods proposed more precise gradient estimation algorithms. For example, Menon *et al*. [9] proposed demosaicking with directional filtering and a posteriori decision using color gradients over a window. In [10], Leung *et al.* tried to solve the demosaicking problem in the frequency domain. Chen and Chang [11] calculated the weight for interpolation by detecting the edge characteristics of a digital image, and Dengwen *et al*. [12] proposed color demosaicking algorithm using directional filtering and weighting. Pekkucuksen *et al*. proposed orientation-free edge strength filter (ESF) [13] and iteratively refined the green channel using neighboring color difference values. They also proposed multi-scale gradient (MSG) [14] based CFA interpolation algorithm which achieves objective performance improvement. In [15], Chen *et al*.

Y. Kim and J.Jeong are with Department of Electronics and Computer Engineering, Hanyang University, Haengdangdong, Sungdonggu, Seoul, South Korea (e-mail: hoonykim85@gmail.com, jjeong@hanyang.ac.kr*).
"This research was supported by the MSIP(Ministry of Science, ICT & Future Planning), Korea, under the project for technical development of information communication & broadcasting supervised by IITP (Institute for Information & Communications Technology Promotion)(IITP 2014-044-057-001)

proposed voting strategy to interpolate color component more accurately.

In this paper, we present gradient estimation methods to suppress color interpolation artifacts and introduce hybrid directional weights (HDW) for adaptive directional interpolation. The hybrid weights are calculated using color correlation and intensity differences, and these weights apply to the whole demosaicking process. The remaining of this paper is described as following. In Section II, the proposed hybrid directional weight-based demosaicking algorithm is explained. Experimental results are described in Section III. Conclusion remarks are reported in Section IV.

II. Proposed Algorithm

The gradient information is important for adaptive directional interpolation. The proposed algorithm uses gradient information which is calculated from the color difference value. The estimated horizontal (H) and vertical (V) color difference are combined with the ratio of integrated gradient of each direction over the window. The color differences are calculated as follows:

$$d_{i,j}^{H}=\begin{cases} Z_{i,j}-Z_{i,j}^{H}, & Z_{i,j}\in\{G\} \\ Z_{i,j}^{H}-Z_{i,j}, & Z_{i,j}\in\{R,B\} \end{cases},\qquad d_{i,j}^{V}=\begin{cases} Z_{i,j}-Z_{i,j}^{V}, & Z_{i,j}\in\{G\} \\ Z_{i,j}^{V}-Z_{i,j}, & Z_{i,j}\in\{R,B\} \end{cases}, \tag{1}$$

where d^H and d^V represent horizontal and vertical color difference estimates, respectively. Parameters Z^H and Z^V are defined as follows:

$$Z_{i,j}^{H}=\frac{2Z_{i,j}-Z_{i,j-2}-Z_{i,j+2}}{4}+\frac{Z_{i,j-3}+9\left(Z_{i,j-1}+Z_{i,j+1}\right)+Z_{i,j+3}}{20},$$
$$Z_{i,j}^{V}=\frac{2Z_{i,j}-Z_{i,j-2}-Z_{i+2,j}}{4}+\frac{Z_{i-3,j}+9\left(Z_{i-1,j}+Z_{i+1,j}\right)+Z_{i+3,j}}{20}. \tag{2}$$

Here, Z denotes the Bayer pattern image plane. Using these values, the gradients are calculated as follows:

$$\Delta_{i,j}^{H}=\left|d_{i,j+1}^{H}-d_{i,j-1}^{H}\right|,\qquad \Delta_{i,j}^{V}=\left|d_{i+1,j}^{V}-d_{i-1,j}^{V}\right|, \tag{3}$$

where Δ^H and Δ^V represent horizontal and vertical gradient. Gradient maps of each direction are produced during this process.

A. Hybrid Directional Weights (HDW)

For the directional interpolation, two sets of weights are used. The first set is vertical and horizontal weights for the initial green channel and R/B interpolation. The second set is four directional weights, up (u), down (d), left (l), and right (r), for green channel refinement step and R/B interpolation. To increase the accuracy, not only the gradient values of color difference but also gradient of pixel intensity of same color plane are exploited. This is because color difference is good measure for estimating gradient while it cannot fully cover the intensity changes. The gradient of same color plane is defined as follows:

$$\Phi_{i,j}^{H}=\left|Z_{i,j+1}-Z_{i,j-1}\right|,\qquad \Phi_{i,j}^{V}=\left|Z_{i+1,j}-Z_{i-1,j}\right|, \tag{4}$$

where Φ^H and Φ^V denote the horizontal and vertical gradients of same color plane, respectively. For the directional weight, Δ^H and Δ^V are used as main factors with window size 5x5, and Φ^H and Φ^V are used as control parameter with window size 3x3. The hybrid directional weights are given as follows:

$$w_{i,j}^{H}=1/\left[\left(\sum_{m=-2}^{2}\sum_{l=-2}^{2}\Delta_{i+m,j+l}^{H}\times\sum_{m=-1}^{1}\sum_{l=-1}^{1}\Phi_{i+m,j+l}^{H}\right)^{2}+1\right],$$
$$w_{i,j}^{V}=1/\left[\left(\sum_{m=-2}^{2}\sum_{l=-2}^{2}\Delta_{i+m,j+l}^{V}\times\sum_{m=-1}^{1}\sum_{l=-1}^{1}\Phi_{i+m,j+l}^{V}\right)^{2}+1\right], \tag{5}$$

where w^H and w^V denote the horizontal and vertical weights and 1 is added to void division by zero. The four hybrid directional weights are calculated similarly. The four weights are calculated as follows:

$$w_{i,j}^{u}=1/\left[\sum_{m=-2}^{0}\sum_{l=-1}^{1}\Delta_{i+m,j+l}^{V}\times\sum_{m=-2}^{0}\sum_{l=-1}^{1}\Phi_{i+m,j+l}^{V}+1\right],$$
$$w_{i,j}^{d}=1/\left[\sum_{m=0}^{2}\sum_{l=-1}^{1}\Delta_{i+m,j+l}^{V}\times\sum_{m=0}^{2}\sum_{l=-1}^{1}\Phi_{i+m,j+l}^{V}+1\right],$$
$$w_{i,j}^{l}=1/\left[\sum_{m=-1}^{1}\sum_{l=-2}^{0}\Delta_{i+m,j+l}^{H}\times\sum_{m=-1}^{1}\sum_{l=-2}^{0}\Phi_{i+m,j+l}^{H}+1\right],$$
$$w_{i,j}^{r}=1/\left[\sum_{m=-1}^{1}\sum_{l=0}^{2}\Delta_{i+m,j+l}^{H}\times\sum_{m=-1}^{1}\sum_{l=0}^{2}\Phi_{i+m,j+l}^{H}+1\right], \tag{6}$$

where w^u, w^d, w^l, and w^r denote weight of up, down, left, and right, respectively.

B. Green Channel Interpolation

In this stage, color difference value which is located at the missing green pixel position is reconstructed by blending directional color difference calculated using Eq. (1). The initial green channel interpolation process is given as follows:

$$\tilde{d}_{i,j} = \frac{d^{H}_{i,j} \cdot w^{H}_{i,j} + d^{V}_{i,j} \cdot w^{V}_{i,j}}{w^{H}_{i,j} + w^{V}_{i,j}}, \tag{7}$$

where $\tilde{d}$ denotes estimated color difference value. In this stage, green value can be calculated from estimated color difference, but estimated values are kept for the next process.

C. Green Channel Update

After the initial green channel interpolation step, color difference estimates are fully reconstructed and they can be further refined with neighboring correlations based on constant color difference assumption. The four close neighbors are combined with estimate color difference with its own weights. The updating process is given as follows:

$$\hat{d}_{i,j} = \tilde{d}_{i,j} \cdot \varepsilon + \frac{\left[w^{u}_{i,j} \cdot \tilde{d}_{i-2,j} + w^{d}_{i,j} \cdot \tilde{d}_{i+2,j} + w^{l}_{i,j} \cdot \tilde{d}_{i,j-2} + w^{r}_{i,j} \cdot \tilde{d}_{i,j+2} \right]}{w^{u}_{i,j} + w^{d}_{i,j} + w^{l}_{i,j} + w^{r}_{i,j}} \cdot (1-\varepsilon), \tag{8}$$

where parameter $\hat{d}$ represents the refined color difference value, and ε is the parameter to control the contribution of original and update part. Parameter ε is obtained empirically, which is given as 0.4 in this paper. This process improves PSNR and reduces the color artifacts, and it performs only once. After complete the process, final green value is calculated as follows:

$$G'_{i,j} = Z_{i,j} + \hat{d}_{i,j}, \tag{9}$$

where G' denotes the interpolated green value.

D. Red and Blue Channel Interpolation

The R/B channel interpolation consists of two steps. Red pixels at the blue pixel location and blue pixels at the red pixel location are interpolated first, then red and blue pixels at the green pixel location are performed. The R/B pixels at the B/R pixel location are estimated by using the 7 by 7 filter. For better estimation with avoiding heavy complexity, the directional weights defined in Eq. (6) are reused to make four diagonal weights. The interpolation filter is given as follows:

$$f^{RB}_{(i,j)} = \frac{1}{6(w^{ul}_{i,j} + w^{ur}_{i,j} + w^{dl}_{i,j} + w^{dr}_{i,j})} \times \begin{bmatrix} 0 & 0 & -w^{ul}_{i,j} & 0 & -w^{ur}_{i,j} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -w^{ul}_{i,j} & 0 & 8 \cdot w^{ul}_{i,j} & 0 & 8 \cdot w^{ur}_{i,j} & 0 & -w^{ur}_{i,j} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -w^{dl}_{i,j} & 0 & 8 \cdot w^{dl}_{i,j} & 0 & 8 \cdot w^{dr}_{i,j} & 0 & -w^{dr}_{i,j} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -w^{dl}_{i,j} & 0 & -w^{dr}_{i,j} & 0 & 0 \end{bmatrix}, \tag{10}$$

where $w^{ul}_{i,j}, w^{ur}_{i,j}, w^{dl}_{i,j}$, and $w^{dr}_{i,j}$ are computed as follows:

$$\begin{aligned} w^{ul}_{i.j} &= w^{u}_{i.j} + w^{l}_{i.j}, \\ w^{ur}_{i.j} &= w^{u}_{i.j} + w^{r}_{i.j}, \\ w^{dl}_{i.j} &= w^{d}_{i.j} + w^{l}_{i.j}, \\ w^{dr}_{i.j} &= w^{d}_{i.j} + w^{r}_{i.j}. \end{aligned} \tag{11}$$

The red and blue pixel estimates at blue and red pixel location is calculated as follows:

$$\begin{aligned} R'_{i.j} &= G'_{i.j} - (\hat{d}_{i.j} \otimes f^{RB}_{i.j}), \\ B'_{i.j} &= G'_{i.j} - (\hat{d}_{i.j} \otimes f^{RB}_{i.j}), \end{aligned} \tag{12}$$

where $\otimes$ denotes the sum of all elements after element-wise matrix multiplication.

The red and blue pixels at green pixel location are interpolated only using weight calculated in Eq. (5). The reason that four hybrid directional weights are not used for this interpolation step is that it does not provide any performance gain. The red and blue pixel estimation is calculated as follows:

$$\begin{aligned} R'_{i,j} &= G'_{i,j} - \left[\frac{w^{V}_{i,j} \cdot \left(d^{GR}_{i-1,j} + d^{GR}_{i+1,j} \right) + w^{H}_{i,j} \cdot \left(d^{GR}_{i,j-1} + d^{GR}_{i,j+1} \right)}{2(w^{V}_{i,j} + w^{H}_{i,j})} \right], \\ B'_{i,j} &= G'_{i,j} - \left[\frac{w^{V}_{i,j} \cdot \left(d^{GB}_{i-1,j} + d^{GB}_{i+1,j} \right) + w^{H}_{i,j} \cdot \left(d^{GB}_{i,j-1} + d^{GB}_{i,j+1} \right)}{2(w^{V}_{i,j} + w^{H}_{i,j})} \right], \end{aligned} \tag{13}$$

where d^{GB} and d^{GR} represent green-blue color difference and green-red color difference.When you submit your final version, after your paper has been accepted, prepare it in two-column format, including figures and tables.

III. EXPERIMENTAL RESULTS

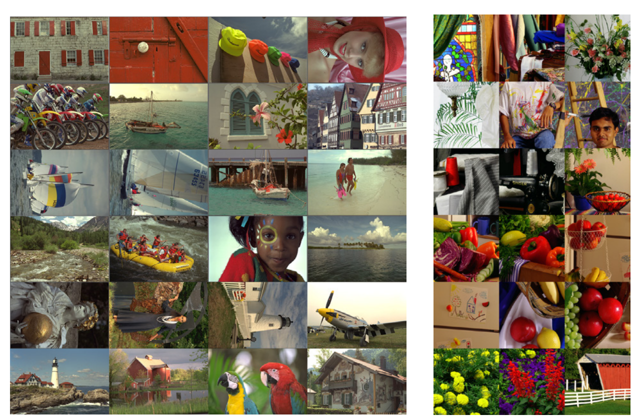

(a) (b)

Fig. 2. (a) Kodak and (b) McM test images.

TABLE I

CPSNR Comparison for Different Demosaicking Algorithms (dB)

Test set	No.	DLMMSE	LSLCD	ESF	EDAEP	MSG	VDI	Proposed
Kodak	1	38.52	38.68	39.91	34.91	39.79	35.34	39.36
	2	40.93	40.81	40.82	39.70	41.59	39.64	41.92
	3	42.75	42.42	42.55	41.69	43.63	41.67	43.90
	4	41.09	41.37	40.45	39.99	41.42	39.56	41.84
	5	38.10	38.38	37.55	35.77	38.94	36.70	39.31
	6	40.27	40.47	41.22	36.83	41.15	37.02	41.15
	7	42.39	42.89	42.15	41.23	43.39	41.86	43.77
	8	36.08	35.75	37.19	32.82	37.39	34.07	37.20
	9	42.86	42.87	42.96	40.85	43.71	41.72	43.83
	10	42.61	42.85	42.60	40.93	43.30	41.25	43.42
	11	40.09	40.35	40.69	37.45	41.23	37.74	41.19
	12	43.53	43.48	43.80	41.61	44.32	41.70	44.55
	13	34.81	35.17	36.11	31.48	35.96	31.35	35.58
	14	37.03	36.99	36.11	36.25	37.77	36.58	38.35
	15	39.87	40.08	39.27	38.78	40.22	38.08	40.46
	16	43.83	44.12	44.77	40.40	44.58	40.62	44.70
	17	41.86	42.04	41.95	39.59	42.46	39.81	42.44
	18	37.45	37.80	37.72	35.37	38.22	35.18	38.21
	19	40.90	40.89	41.49	37.73	41.99	38.97	41.90
	20	41.27	41.28	41.52	39.24	42.13	39.96	42.14
	21	39.17	39.33	40.30	36.43	40.26	37.03	40.00
	22	38.46	38.70	38.41	37.15	39.00	37.46	39.14
	23	43.30	43.16	42.45	42.29	43.89	41.42	44.16
	24	35.52	35.59	35.34	33.72	35.72	33.07	35.67
Average		40.11	40.23	40.31	38.01	40.92	38.24	41.01
McM	1	26.98	26.05	26.38	27.60	27.05	28.01	27.61
	2	33.68	33.05	33.48	33.99	33.67	34.18	34.12
	3	32.59	32.34	32.56	32.07	32.93	32.64	33.24
	4	34.32	35.00	34.97	34.36	35.49	36.00	36.09
	5	31.27	30.61	30.64	32.10	31.12	32.63	31.89
	6	33.84	33.08	32.57	35.03	33.56	35.63	34.59
	7	38.64	38.33	39.10	36.22	39.17	36.03	38.77
	8	37.45	36.70	37.85	37.12	37.61	37.41	38.24
	9	34.41	33.67	34.39	35.23	34.69	35.95	35.51
	10	36.34	35.48	35.78	37.01	36.47	37.28	37.08
	11	37.25	36.22	36.61	37.83	37.28	37.98	37.86
	12	36.60	36.12	36.16	37.16	36.80	37.09	37.33
	13	38.79	38.21	38.67	39.34	38.83	39.40	39.45
	14	37.23	36.52	37.21	37.65	37.13	37.32	37.56
	15	37.27	36.55	37.01	37.76	37.19	37.86	37.62
	16	30.46	29.05	29.24	31.39	30.18	31.40	30.85
	17	29.31	28.65	28.57	30.59	29.30	31.15	30.20
	18	33.92	33.05	33.69	34.05	34.10	34.23	34.50
Average		34.46	33.82	34.16	34.81	34.59	35.12	35.14
Total average		37.69	37.48	37.67	36.64	38.21	36.90	38.49

To evaluate the performance of the proposed HDW algorithm, it is tested on the Kodak and McM of color test-images shown in Fig. 2. The Kodak images consist of 24 images with size of 768x512 and McM consist of 18 images with size of 500x500. The proposed HDW has been compared with directional linear minimum mean square-error estimation (DLMMSE), least-squares luma-chorma demultiplexing (LSLCD), effective demosaicking algorithm based on edge property (EDAEP), edge strength filter (ESF), multi-scale gradient (MSG), and voting-based directional interpolation (VDI). The interpolated images with various methods are compared excluding the border within 10 pixels, and we conducted simulations using MATLAB with an Intel(R) Core(TM) i7-4770K CPU at 3.50-GHZ quad-core processor.

The CPSNR results are presented in Table I for each 42 images. In case of Kodak image set, the proposed HDW shows the best average CPSNR among all algorithms and its average CPSNR was 0.09dB and 1.77dB higher than MSG and VDI, respectively. The proposed algorithm also shows the best

ISBN: 978-1-61804-290-3

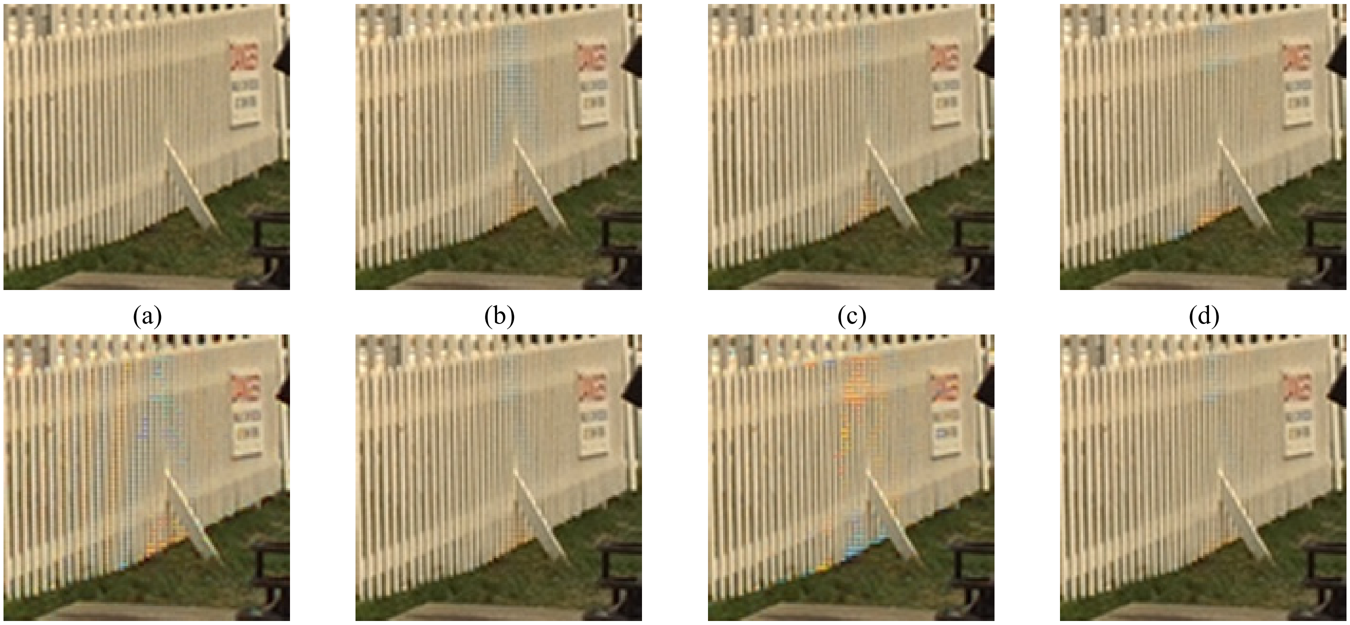
TABLE II

S-CIELAB ΔE^* Comparison for Different Demosaicking Algorithms

Test set	No.	DLMMSE	LSLCD	ESF	EDAEP	MSG	VDI	Proposed
Kodak	1	1.108	1.133	0.995	1.474	0.960	1.368	0.969
	2	0.649	0.642	0.663	0.703	0.601	0.728	0.582
	3	0.479	0.508	0.510	0.524	0.450	0.541	0.439
	4	0.672	0.643	0.727	0.729	0.639	0.795	0.614
	5	1.070	1.011	1.242	1.271	1.011	1.107	0.902
	6	0.774	0.766	0.741	0.989	0.727	0.950	0.689
	7	0.574	0.523	0.584	0.664	0.489	0.564	0.479
	8	1.332	1.443	1.190	1.825	1.138	1.477	1.118
	9	0.569	0.581	0.585	0.653	0.534	0.624	0.520
	10	0.544	0.541	0.569	0.610	0.518	0.617	0.506
	11	0.737	0.730	0.722	0.923	0.670	0.859	0.649
	12	0.461	0.478	0.466	0.550	0.432	0.529	0.422
	13	1.560	1.539	1.477	2.012	1.448	2.042	1.422
	14	0.981	0.971	1.043	1.138	0.871	1.091	0.836
	15	0.657	0.633	0.722	0.691	0.636	0.726	0.610
	16	0.531	0.525	0.508	0.704	0.509	0.689	0.490
	17	0.517	0.506	0.543	0.641	0.499	0.618	0.483
	18	1.008	0.964	1.092	1.160	0.969	1.147	0.930
	19	0.752	0.753	0.736	0.957	0.674	0.847	0.669
	20	0.545	0.564	0.555	0.647	0.502	0.579	0.497
	21	0.901	0.889	0.861	1.091	0.831	1.019	0.812
	22	0.964	0.907	1.023	1.056	0.918	0.995	0.889
	23	0.495	0.479	0.560	0.536	0.465	0.544	0.457
	24	1.018	0.969	1.102	1.156	1.019	1.118	0.970
Average		0.787	0.779	0.801	0.946	0.730	0.899	0.706
McM	1	3.204	3.493	3.784	2.903	3.170	2.620	2.922
	2	1.315	1.295	1.453	1.246	1.252	1.144	1.152
	3	2.083	1.942	2.076	2.290	1.825	1.893	1.784
	4	1.443	1.236	1.461	1.484	1.086	1.113	1.021
	5	1.615	1.714	1.771	1.526	1.551	1.374	1.430
	6	1.511	1.594	1.857	1.278	1.502	1.127	1.311
	7	0.941	1.001	0.944	1.228	0.897	1.199	0.942
	8	0.631	0.637	0.659	0.723	0.589	0.613	0.516
	9	1.377	1.313	1.396	1.305	1.203	1.058	1.084
	10	1.083	1.075	1.192	0.999	1.025	0.932	0.940
	11	0.793	0.827	0.907	0.725	0.783	0.675	0.725
	12	1.089	1.098	1.257	0.940	1.081	0.956	0.993
	13	0.711	0.783	0.739	0.683	0.692	0.650	0.658
	14	0.797	0.807	0.860	0.789	0.784	0.742	0.749
	15	0.855	0.848	0.913	0.819	0.836	0.773	0.795
	16	2.064	2.572	2.839	1.673	2.317	1.499	2.079
	17	2.826	2.647	3.090	2.290	2.577	1.960	2.219
	18	1.510	1.582	1.601	1.451	1.460	1.287	1.358
Average		1.436	1.470	1.600	1.353	1.368	1.201	1.260
Total average		1.065	1.075	1.143	1.120	1.003	1.028	0.944

performance on McM datasets, and the average CPSNR of proposed algorithm is better than VDI by 0.02dB and EDAEP by 0.33dB. Interestingly, most algorithms work well only for one dataset, but do not for the other dataset. For the Kodak images, DLMMSE, LSLCD, ESF, and MSG show fine results but they gave relatively worse results on McM images. On the other hand, EDAEP and VDI only perform well on McM datasets. It is remarkable that the proposed algorithm gives the best performance on both datasets. In terms of total average CPSNR, HDW was better than the MSG, which shows the second best results on Kodak dataset, by 0.28dB, and VDI, which gives second best results on McM dataset, by 1.59dB.

To further evaluate the objective performance, we exploited another metric S-CIELAB ΔE^*, which measures how accurate the reproduction of a color is to the original when viewed by a human observer, and the results are provided in Table II. It can be seen that the proposed algorithm outperforms other algorithms in terms of total average S-CIELAB ΔE^*. MSG



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 3. Picket fence region from the #19 Kodak image. (a) Original, (b) DLMMSE, (c) LSLCD, (d) ESF, (e) EDAEP, (f) MSG, (g) VDI, and (h) HDW (proposed).

TABLE III

Average Computational Time Comparison (Tested on Images with Size of 768x512)

	DLMMSE	**LSLCD**	**ESF**	**EDAEP**	**MSG**	**VDI**	**Proposed (HDW)**
Time (s)	12.89	4.52	9.31	0.46	8.29	5.31	3.09

scores the second best performance followed by VDI. The VDI shows fine results in terms of total average of S-CIELAB ΔE^* even it gives the second worst total average CPSNR.

The visual comparison of the picket fence region of the lighthouse (#19 in Kodak dataset) is provided in Fig. 3. It can be seen that DLMMSE, VDI and EDAEP show severe rainbow artifacts at the high frequency region. The proposed algorithm gives the clear image and successfully removes the most of the demosaicking artifact. In Fig. 4, demosaicking results of partial zoomed image (#18 in McM dataset) are presented. It is obvious that EDAEP and VDI produce more artifacts around the character and edge of the roof, and DLMMSE, LSLCD, and ESF show zipper effect at the region where chrominance component abruptly changes across the boundary.

In Table III, the computational complexity of HDW and the other demosaicking methods were evaluated using computational time. The execution times have been evaluated using the source codes which were originally distributed from the authors under the same condition. To evaluate computational complexity, the Kodak dataset was used to calculate the computational time. From the results, we can see that the proposed algorithm gives approximately 1.6 times faster than VDI which gives the second best results on McM dataset and 2.6 times faster than MSG which shows the second best results on Kodak dataset.

From the above objective and subjective evaluations results, it can be seen that the proposed algorithm outperformances other state-of-the-art algorithms with low complexity burden.

IV. Conclusion

In this paper, a hybrid directional weight for CFA interpolation algorithm is proposed, which exploits both gradient of color difference and intensity difference for calculating weights. The whole interpolation process is adaptively combined with hybrid directional weights and it leads to reducing the common demosaicking artifacts and improving the interpolation performance in terms of average CPSNR, S-CIELAB ΔE^{*}.

References

[1] B. E. Bayer, “Color imaging array,” U.S. Patent 3 971 065, Jul. 1976.

[2] R. Kimmel, “Demosaicing: Image reconstruction from color CCD samples,” *IEEE Trans. Image Process*, vol. 8, no. 9, pp. 1221–1228, Sep. 1999.

[3] J. F. Hamilton and J. E. Adams, “Adaptive color plane interpolation in single sensor color electronic camera,” U.S. Patent 5 629 734, May 1997.

[4] C. A. Laroche andM. A. Prescott, “Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients,” U.S. Patent 5 373 322, Dec. 1994.

[5] Y.M. Lu, M. Karzand, M. Vetterli, ‘Demosaicking by alternating projections: theory and fast one-step implementation’, *IEEE Trans. Image Process*., vol. 19, no. 8, pp. 2085–2098, Aug. 2010.

[6] K. H. Chung and Y. H. Chan, "Color demosaicing using variance of color differences," *IEEE Trans. Image Process.,* vol. 15, no. 10, pp. 2944–2955, Oct. 2006.

[7] D. Paliy, V. Katkovnik, R. Bilcu, S. Alenius, and K. Egiazarian, "Spatially adaptive color filter array interpolation for noiseless and noisy data," *Int. J. Imag. Syst. Technol.,* vol. 17, no. 3, pp. 105–122, 2007.

[8] L. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-error estimation," *IEEE Trans. Image Process.,* vol. 14, no. 12, pp. 2167–2178, Dec. 2005.

[9] D. Menon, S. Andriani, and G. Calvagno, "Demosaicing with directional filtering and a posteriori decision," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 132–141, Jan. 2007.

[10] B. Leung, G. Jeon and E. Dubois, "Least-squares luma-chroma demultiplexing algorithm for Bayer demosaicking," *IEEE Trans. Image Process*., vol. 20, no. 7, pp. 1885–1894, Jul. 2011.

[11] W. Chen, P. Chang, "Effective demosaicking algorithm based on edge property for color filter arrays," *Digit. Signal Process.,* vol. 22, no. 1, pp. 163–169, Jan. 2012.

[12] Z. Dengwen, S. Xiaoliu and D. Weiming, "Colour demosaicking with directional filtering and weighting," *IET Image Process.,* vol. 6, no. 8, pp. 1084–1092, Nov. 2012.

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

[14] I. Pekkucuksen and Y. Altunbasak, "Multiscale Gradients-Based Color Filter Array Interpolation," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 157–165, Jan. 2013.

[15] X. Chen, G. Jeon, and J. Jeong, "Voting-based directional interpolation method and its application to still color image demosaicking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 255-262, Feb. 2014.

[16] L. Zhang, L, Zhang, X. Mou, and D. Zhang, "FSIM: a feature similarity index for image quality assessment," *IEEE Trans. Image Process.,* vol. 20, no. 8 pp. 2378–2386, Aug. 2011.