ATLANTIS PRESS

Advances in Computer Science Research, (ACSR), volume 52
2016 International Conference on Computer Engineering and Information Systems (CEIS-16)

Artifacts Reduced Interpolation Method for Single-Sensor Imaging System

Long-Fei Wang
College of Telecommunications & Information Engineering,
Nanjing University of Posts and Telecommunications,
Nanjing, 210046, China
Email: wanglf9542@163.com

Si-Chen Zhou, Xin-Yi Peng
School of Oversea
Nanjing University of Posts and Telecommunications
Nanjing, 210046, China

Xiang-Dong Chen
School of Computer Science,
Nanjing University of Posts and Telecommunications,
Nanjing, 210046, China
Email: chenxd@njupt.edu.cn

***Abstract*-In this paper, we present a novel color image Demosaicking algorithm. The algorithm consists of two steps: an interpolation step and a refinement step. The missing green color information is first interpolated by using the color channel difference. In the refinement step, a local weighted directional interpolation method guided by the pre-interpolated green channel is applied to refine the interpolation results along the determined interpolation direction. Lastly, post-processing is implemented to output the final Demosaicked full color image. Compared with the latest Demosaicking algorithms, experiments showed that the proposed method provides superior performance in terms of both objective and subjective image qualities.**

Keywords-demosaicking; CFA interpolation; artifacts reduction

I. INTRODUCTION

When currently available digital still color cameras based on a single charge-coupled device (CCD) sensor capture a color pixel, only one part of the color information of the three color channels is captured. To reconstruct a full-color image, an interpolation process, commonly referred to CFA interpolation, is applied to estimate the other two missing color pixel values at each pixel position. This process is called CFA interpolation, or Demosaicking. Presently, the most common CFA in digital cameras uses a color arrangement based on the Bayer pattern [1, 2]. Fig. 1 shows a 7×7 window of Bayer CFA samples.

The color reproduction quality depends on the CFA templates and the employed Demosaicking algorithms. Various Demosaicking algorithms based on the Bayer pattern [3-13] have been proposed in the past decades. Recently developed methods include the successive approximation (SA) method by Li [4], the directional linear minimum mean square-error estimation (DL) method by Zhang and Wu [5], a least-squares luma-chroma demultiplexing (LSLCD) algorithm for Bayer Demosaicking by Dubois et al. [6], an effective Demosaicking method based on edge property (EDEP) by Chen and Chang [7], an adaptive filtering for color filter array Demosaicking (AFD) in the frequency domain proposed by Lian et al. [8], and the edge strength filter (ESF) based method by Pekkucuksen and Altunbasak [9]. A recent survey of Demosaicking methods can be found in [14]. Some of these methods exploit intra channel correlation (the color difference from green-to-green, red-to-red, and blue-to-blue) to determine the interpolation while others use the inter channel correlation (the color difference from green-to-red, green-to-blue, and red-to-blue). In the literature, methods use inter channel correlation have yielded better performance.

In this paper, we present a new color image Demosaicking algorithm. We first utilize the color difference between channels to populate the green (G) channel in advance, then a local weighted directional interpolation method is used to refine the green channel. The pre-interpolated green channel is used to calculate the directional gradient since it supports more accurate edge information than conventional methods. These directional gradients in the working window are used to determine the interpolation direction. The pre-interpolation result is refined along the determined interpolation direction. Finally, we apply a post-processing approach to remove interpolation artifacts by utilizing the directional weighted mean of neighboring color differences over channels.

The remainder of the paper is organized as follows. The proposed method including green channel interpolation and refinement, red (R) and blue (B) component interpolation, and overall plane refinements are described in Section II. We evaluate the Demosaicking performance of the conventional and the proposed methods in Section III. Finally, conclusions are made in Section IV.

II. THE PROPOSED METHOD

The green plane is usually reconstructed first because it contains twice as many samples as the red or blue planes. Thus, the green plane possesses most of the spatial information of the image to be interpolated and has great influence on the perceptual quality of the image.

Furthermore, once the green plane is fully populated, the green plane can be used to guide the subsequent red and blue plane interpolation by making full and direct use of channel correlation.

A. *Green Channel Interpolation*

The green channel is interpolated in two steps. In the first step, we use the color differences κ_R (=G–R) and κ_B (=G–B). The green channel can be roughly interpolated in four directions: north (N), south (S), west (W), and east (E). The inter-channel correlation is exploited as the weighted factor to adjust the contribution of color differences among neighboring pixels. As illustrated in Fig. 1, the green pixel value at a location R_5 can be obtained by first calculating the κ_R values of the four points surrounding R_5, that is, G_9, G_{16}, G_{12}, and G_{13}. The κ_R values of the four points are calculated by the following equations, respectively:

$$\begin{aligned}\kappa_{R_9} &= G_9 - \frac{1}{2}(R_2 + R_5)\\ \kappa_{R_{16}} &= G_{16} - \frac{1}{2}(R_8 + R_5)\\ \kappa_{R_{12}} &= G_{12} - \frac{1}{2}(R_4 + R_5)\\ \kappa_{R_{13}} &= G_{13} - \frac{1}{2}(R_6 + R_5)\end{aligned} \tag{1}$$

Next, the absolute color gradients of the channels that measure the spatial correlations of the neighboring pixels G_9, G_{16}, G_{12}, and G_{13} along the four directions are calculated as:

$$\begin{aligned}D_X^W &= \sum_{i=a-1}^{a+1}\sum_{j=b-2}^{b-1} |X(i,j) - X(i,j+1)|\\ D_X^E &= \sum_{i=a-1}^{i=a+1}\sum_{j=b}^{j=b+1} |X(i,j) - X(i,j+1)|\\ D_X^N &= \sum_{i=a-2}^{a-1}\sum_{j=b-1}^{b+1} |X(i,j) - X(i+1,j)|\\ D_X^S &= \sum_{i=a}^{a+1}\sum_{j=b-1}^{b+1} |X(i,j) - X(i+1,j)|\end{aligned} \tag{2}$$

where $X(i, j)$ is Bayer-patterned CFA at position (i, j), and (a, b) is the position of the central missing color component in the local sliding window. D_X^W and D_X^E are the absolute color gradients of G_{12} and G_{13} in the horizontal direction, and D_X^N and D_X^S are the absolute color gradients of G_9 and G_{16} in the vertical direction.

The inverse items of absolute color difference are used as weight factors to adjust for the contribution of each κR value according to their spectral correlation with the central missing color component. The weight allocated to each κR is listed as follows:

$$\begin{aligned}w_{R5}^W &= \frac{1}{D_X^W}, \quad w_{R5}^E = \frac{1}{D_X^E}\\ w_{R5}^N &= \frac{1}{D_X^N}, \quad w_{R5}^S = \frac{1}{D_X^S}\end{aligned} \tag{3}$$

Then, the generalized color difference κ_{R5} of the central missing green component and R_5 are estimated using the surrounding κ_R value and its corresponding weight by the following equation:

$$\begin{aligned}\kappa_{R5} = (&\kappa_{R9} * w_{R5}^N + \kappa_{R16} * w_{R5}^S + \kappa_{R12} * w_{R5}^W\\ &+\kappa_{R13} * w_{R5}^E)/(w_{R5}^N + w_{R5}^S + w_{R5}^W\\ &+ w_{R5}^E)\end{aligned} \tag{4}$$

Finally, the missing green component $\widetilde{G}_{R5}$ is interpolated at the position of R5 as:

$$\widetilde{G}_{R5} = R_5 + \kappa_{R5} \tag{5}$$

B. *Green Channel Refinement*

Once the missing green component is populated, it can be used to determine the interpolation direction in the refinement step. In the refinement step, every pre-interpolated color component is refined by combining the estimates obtained from its four interpolation directions by exploiting the spectral correlation among the neighboring pixels along that direction. Utilizing the color difference between the R and G channels, $\widetilde{G}_{R5}$ can be estimated along the four directions. Referring to Fig. 1, $\widetilde{G}_{R5}$ is estimated as $\widetilde{G}_{R5}^N$, $\widetilde{G}_{R5}^S$, $\widetilde{G}_{R5}^W$, and $\widetilde{G}_{R5}^E$ in these four directions as:

$$\begin{aligned}\widetilde{G}_{R5}^N &= G_9 + 0.5 * (R_5 - R_2)\\ \widetilde{G}_{R5}^S &= G_{16} + 0.5 * (R_5 - R_8)\\ \widetilde{G}_{R5}^W &= G_{12} + 0.5 * (R_5 - R_4)\\ \widetilde{G}_{R5}^E &= G_{13} + 0.5 * (R_5 - R_6)\end{aligned} \tag{6}$$

For better estimation of $\widetilde{G}_{R5}$, we assign each estimate with an appropriate weight using the pre-interpolated green channel, and the directional gradients of R5 along the four directions are calculated by:

$$\begin{aligned}\Delta^N &= |G_5 - \tilde{\tilde{G}}_{B6}| + |\tilde{\tilde{G}}_{R2} - G_9| + |G_6 - \tilde{\tilde{G}}_{B7}|\\ &\quad +|\tilde{\tilde{G}}_{B6} - G_{12}| + |G_9 - \tilde{\tilde{G}}_{R5}| + |\tilde{\tilde{G}}_{B7} - G_{13}| + \varepsilon\\ \Delta^S &= |G_{12} - \tilde{\tilde{G}}_{B10}| + |\tilde{\tilde{G}}_{R5} - G_{16}| + |G_{13} - \tilde{\tilde{G}}_{B11}|\\ &\quad +|\tilde{\tilde{G}}_{B10} - G_{19}| + |G_{16} - \tilde{\tilde{G}}_{R8}| + |\tilde{\tilde{G}}_{B11} - G_{20}| + \varepsilon\\ \Delta^W &= |G_8 - \tilde{\tilde{G}}_{B6}| + |\tilde{\tilde{G}}_{R4} - G_{12}| + |G_{15} - \tilde{\tilde{G}}_{B10}|\\ &\quad +|\tilde{\tilde{G}}_{B6} - G_9| + |G_{12} - \tilde{\tilde{G}}_{R5}| + |\tilde{\tilde{G}}_{B10} - G_{16}| + \varepsilon\\ \Delta^E &= |G_9 - \tilde{\tilde{G}}_{B7}| + |\tilde{\tilde{G}}_{R5} - G_{13}| + |G_{16} - \tilde{\tilde{G}}_{B11}|\\ &\quad +|\tilde{\tilde{G}}_{B7} - G_{10}| + |G_{13} - \tilde{\tilde{G}}_{R6}| + |\tilde{\tilde{G}}_{B11} - G_{17}| + \varepsilon\end{aligned} \tag{7}$$

Where, ε is a small positive factor to avoid the gradient being zero. The interpolation direction of R_5 is determined by the directional gradients according to the distribution situation of the four directional gradients of R_5: Δ^N, Δ^S, Δ^W, and Δ^E. The final interpolation of $\widetilde{G}_{R5}$ can be classified as one of three situations.

If $\alpha(\Delta^W+\Delta^E)<(\Delta^N+\Delta^S)$, the interpolation direction is determined to be horizontal, and the interpolation is only applied in the west and east directions.

If $\alpha(\Delta^N+\Delta^S)<(\Delta^W+\Delta^E)$, the interpolation direction is determined to be vertical, and the interpolation is only applied in the north and south directions.

Otherwise, the interpolation direction is undefined, and the interpolation is applied along all four directions. Here, the coefficient α (α≥1) is used as a constraint factor to judge the interpolation direction.

The inverse of the directional gradients are used as the weight factors to adjust the joint contribution of estimation along the four interpolation directions similarly to the pre-interpolation step. They are represented as:

$$\eta^N=\frac{1}{\Delta^N},\quad \eta^S=\frac{1}{\Delta^S},\quad \eta^W=\frac{1}{\Delta^W}\quad ,\eta^E=\frac{1}{\Delta^E} \tag{8}$$

The interpolation equations are given according to the three determined directions. For horizontal interpolation, the estimations of $\widetilde{G}_5^W$ and $\widetilde{G}_5^E$ in the horizontal direction are used, and the weighting factors of η^W and η^E are involved in order to adjust the interpolation performance. For horizontal interpolation, the normalized interpolation equation is given by:

$$\widetilde{G}_{R5}=\frac{\widetilde{G}_{R5}^W*\eta^W+\widetilde{G}_{R5}^E*\eta^E}{\eta^W+\eta^E} \tag{9}$$

Similarly, for vertical interpolation, the normalized interpolation equation is defined as:

$$\widetilde{G}_{R5}=\frac{\widetilde{G}_{R5}^N*\eta^N+\widetilde{G}_{R5}^S*\eta^S}{\eta^N+\eta^S} \tag{10}$$

For the case where the interpolation direction is undefined, the interpolation is estimated along the four directions in order to avoid interpolation error. In other words, we use the joint contribution of all the pre-estimations $\widetilde{G}_{R5}^N$, $\widetilde{G}_{R5}^S$, $\widetilde{G}_{R5}^W$, and $\widetilde{G}_{R5}^E$ in four directions to guarantee the accuracy of interpolation with the weighting factors of η^N, η^S, η^W, and η^E. The normalized interpolation equation is defined as:

$$\widetilde{G}_{R5}=\frac{\widetilde{G}_{R5}^n*\eta^N+\widetilde{G}_{R5}^s*\eta^S+\widetilde{G}_{R5}^w*\eta^W+\widetilde{G}_{R5}^e*\eta^E}{\eta^N+\eta^S+\eta^W+\eta^E} \tag{11}$$

By applying the above procedures to all red and blue positions, we can refine the green plane.

C. *Interpolating the Missing Red and Blue Components*

From the Bayer CFA samples, the green pixels are initially interpolated by the proposed method. Since the red, green, and blue planes are highly correlated, the interpolation process for R and B uses their color difference planes to avoid color mis-registration problems. First, the color difference planes δ_{RG} and δ_{BG} are calculated by Eq. (12).

$$\delta_{RG}=R-G,\quad \delta_{BG}=B-G \tag{12}$$

Thus, red and blue pixels can be reconstructed by Eq. (13) as follows:

$$R=G+\delta_{RG}\,,\qquad B=G+\delta_{BG} \tag{13}$$

Specifically, the color difference planes are calculated under two conditions: the missing red and blue components at green CFA sampling positions and the missing blue (or red) components at red (or blue) sampling positions. Different neighboring pixels are used to interpolate the missing red and blue pixels according to the position condition.

In order to reduce the interpolation artifacts, a refinement scheme processes the interpolated green samples G first to enhance the interpolation performance, and based on the refined green plane, it performs a refinement of the interpolated red and blue samples. More details on this refinement scheme can be found in [7, 12].

III. EXPERIMENTAL RESULTS

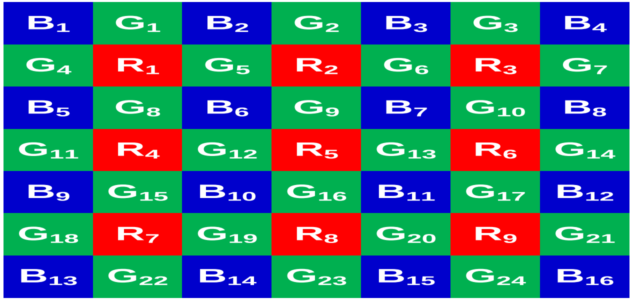
In this section, the proposed local adaptive directional interpolation algorithm (LADI) is evaluated both objectively and subjectively, and compared with various Demosaicking methods. The first 18 digital color images from the Kodak image dataset and were used to generate a set of testing images [15]. To conduct the experiments, we first implemented the mosaicking procedure using a Bayer color filter array on the target testing images, and then applied different Demosaicking methods to reconstruct the whole three-color-channel demosaicked image. Finally, we compared LADI with the DL, LSLCD, EDEP, AFD, and ESF methods. In addition, a refinement non-embedded LADI (labeled LADI_N) was also listed to determine the improvement in the embedded refinement method in LADI. To validate the proposed algorithm we conducted simulations using MATLAB 2009a on a Intel(R) Core(TM) i5 CPU M460 @2.53GHZ processor.

Table I shows the color peak signal-to-noise ratio (CPSNR) for objective comparison. It can be seen from Table I that our proposed method gave the highest average CPSNR value, and ESF and DL were the second and the third best of the compared methods. On the other hand, LSLCD showed the worst objective quality in the comparison. It is obvious that after refinement, LADI has a much higher CPSNR value than LADI_N.

Table II shows the objective image quality with the index of zipper effect ratio (ZER) [13]. In terms of ZER, the proposed LADI method gives the best performance with the least severe zipper effect, followed by ESF, which had the second least serious zipper effect. Although AFD had the lowest ZER value in many test images (the third best performance in average ZER metric), it did not have any advantage in terms of average ZER due to the lack of robustness and reliability for all images. In comparison, it can be intuitively observed that LADI outperformed LADI_N due to the efficiency of the refinement processing.

From the comparisons of the three objective evaluations, LADI showed competitive performance among all the methods tested, and its interpolation of various test images was accurate and robust. It should be noted that all of the measures in our experiments were computed after removing a ten-pixel-wide boundary around the border of the image.

For subjective evaluation, we used images #1, #15 from the Kodak dataset for subjective performance evaluation. Zoomed-in portions of demosaicked images are presented in Figs. 2(a) and Figs.3 (a). In Figs. 2, the counterpart images from the compared Demosaicking methods are shown to demonstrate artifact abilities blocking along the intensive edges of the window shades. The demosaicked image from LADI showed clear edges, just like the original figure. LADI caused the fewest color artifacts compared to other methods, as seen in Fig. 2(i). It is noteworthy that even without refinement, LADI_N caused fewer interpolation artifacts than other methods, which can be seen in Fig. 2(g). A similar comparison of texture-preserving ability using image #15 is shown in Figs. 3. In Fig. 3(a), the windows with blinds have an intensive and texture-like narrow edge. Due to the well-exploited inter-channel correlation, the edge direction is well estimated. Thus, even with this narrow, short edge, our proposed methods LADI_N and LADI, can recover the edge with inconspicuous color artifacts, as shown in Fig. 3(g, h). The other methods showed more or fewer color artifacts and suffered distortions in the edge direction to a variable degree, which can be seen in Figs. 3(b-f).



B_1	G_1	B_2	G_2	B_3	G_3	B_4
G_4	R_1	G_5	R_2	G_6	R_3	G_7
B_5	G_8	B_6	G_9	B_7	G_{10}	B_8
G_{11}	R_4	G_{12}	R_5	G_{13}	R_6	G_{14}
B_9	G_{15}	B_{10}	G_{16}	B_{11}	G_{17}	B_{12}
G_{18}	R_7	G_{19}	R_8	G_{20}	R_9	G_{21}
B_{13}	G_{22}	B_{14}	G_{23}	B_{15}	G_{24}	B_{16}

Figure 1. A 7×7 Bayer CFA block.

IV. CONCLUSIONS

In this paper, we proposed an efficient Demosaicking algorithm that applies a gradient inverse weighted interpolation method along the interpolation direction as determined by the distribution of the directional gradient. The results showed that our method can determine the interpolation direction accurately. By using the refinement method within the same green channel, artifacts can be avoided. Consequently, our proposed interpolation method has advantages for preserving smooth edges and details.

ACKNOWLEDGMENT

This work was sponsored by NUPTSF (Grant No.NY213087).

REFERENCES

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

[2] H. J. Trussell and R. E. Hartwig, “Mathematics for demosaicking,” IEEE Trans. Image Processing, vol. 11, no. 4, pp. 485-492, Apr. 2002.

[3] S. C. Pei and I. K. Tam, “Effective color interpolation in CCD color filter arrays using signal correlation,” IEEE Trans. Circuits and Systems for Video Technology, vol.13, no. 6, pp.503-513, Jun. 2003.

[4] X. Li, “Demosaicing by successive approximation,” IEEE Trans. Image Processing, vol. 14, no. 3, pp. 370-379, Mar. 2005.

[5] L. Zhang and X. Wu, “Color demosaicking via directional linear minimum mean square-error estimation,” IEEE Trans. Image Processing, vol. 14, no. 12, pp. 2167-2178, Dec. 2005.

[6] B. Leung, G. Jeon, and E. Dubois, “Least-squares luma-chroma demultiplexing algorithm for Bayer demosaicking,” IEEE Trans. Image Processing, vol. 20, no. 7, pp. 1885-1894, Jul. 2011.

[7] W. J. Chen and P. Y.Chang, “Effective demosaicing algorithm based on edge property for color filter arrays,” Digital Signal Processing, vol. 22, no. 1, pp. 163-169, 2012.

[8] N. X. Lian, L. Chang, Y. -P. Tan, and V. Zagorodnov, “Adaptive filtering for color filter array demosaicking,” IEEE Trans. Image Processing, vol. 16, no. 10, pp. 2515-2525, Oct. 2007.

[9] I. Pekkucuksen and Y. Altunbasak, “Edge strength filter based color filter array interpolation,” IEEE Trans. Image Processing, vol. 21, no. 1, pp. 393-397, Jan. 2012.

[10] W. Lu and Y. Tan, “Color filter array demosaicking: new method and performance measures,” IEEE Trans. Image Processing, vol. 12, no. 10, pp. 1194-1210, 2003.

[11] K. H. Chung and Y. H. Chan. “Color demosaicking using variance of color differences,” IEEE Trans. Image Processing, vol. 15, no. 10, pp. 2944-2955, Oct. 2006.

[12] C. Y. Tsai and K. T. Song, “A new edge-adaptive demosaicking algorithm for color filter arrays,” Image and Vision Computing, vol. 25, no. 9, pp. 1495-1508, Sept. 2007.

[13] A. Buades, B. Coll, J.-M. Morel, and C. Sbert, “Self-similarity driven color demosaicking,” IEEE Trans.Image Processing, vol. 18, no. 6, pp. 1192-1202, June 2009.

[14] X. Li, B. Gunturk, and L. Zhang, “Image demosaicing: a systematic survey,” in Proc. of SPIE, vol. 6822, pp. 68221J, 2008.

[15] Kodak color image dataset, http://r0k.us/graphics/kodak/

TABLE I. TABLE CPSNR COMPARISON (IN DB) OF DIFFERENT DEMOSAICKING SCHEMES FOR KODAK IMAGE DATASET

Image	**DL**	**LSLCD**	**EDEP**	**AFD**	**ESF**	**LADI$_N$**	**LADI**	**Rank**
1	38.396	39.367	38.434	37.432	**39.801**	37.624	39.548	2
2	**40.849**	39.533	39.976	40.629	40.738	39.857	40.106	4
3	42.552	40.498	42.668	42.516	42.348	42.441	**42.943**	1
4	40.436	39.514	40.448	40.418	39.95	40.528	**40.694**	1
5	37.966	36.788	38.058	37.903	37.418	37.274	**38.468**	1
6	40.104	40.132	39.588	37.859	**41.064**	38.145	40.253	2
7	42.316	40.814	42.268	**42.823**	42.083	42.568	42.651	2
8	35.978	35.744	35.862	35.096	**37.105**	35.424	36.537	2
9	42.972	41.335	42.677	42.615	42.886	42.127	**43.115**	1
10	42.563	41.759	42.522	42.62	42.457	41.864	**42.758**	1
11	39.934	39.602	39.712	39.163	**40.554**	38.379	40.057	2
12	43.368	42.732	43.283	42.586	43.658	42.539	**43.698**	1
13	34.712	35.888	35.009	33.655	**36.008**	34.543	35.49	4
14	36.781	34.364	36.204	**36.923**	35.844	36.784	36.538	4
15	**39.799**	39.149	39.704	39.78	39.186	39.16	39.698	4
16	43.667	43.555	42.933	40.962	**44.226**	42.005	43.954	2
17	41.574	41.204	41.596	41.127	**41.67**	40.207	41.503	4
18	37.777	37.533	37.803	37.405	**37.983**	36.937	37.925	2
Avg.	40.097	39.417	39.930	39.528	40.277	39.356	**40.330**	1

TABLE II. TABLE ZER COMPARISON OF DIFFERENT DEMOSAICKING SCHEMES FOR THE KODAK IMAGE DATASET

Image	**DL**	**LSLCD**	**EDEP**	**AFD**	**ESF**	**LADI$_N$**	**LADI**	**Rank**
1	1.721	2.178	1.807	1.835	**1.351**	1.936	1.481	2
2	1.098	2.634	1.288	1.018	1.248	**1.097**	1.296	6
3	0.605	1.363	0.641	0.594	0.566	0.643	**0.527**	1
4	0.887	1.267	1.032	**0.825**	0.924	1.126	0.971	4
5	1.091	1.918	1.088	0.928	1.006	0.98	**0.898**	1
6	0.942	1.361	1.084	1.244	**0.753**	1.116	0.919	2
7	0.613	1.799	0.719	0.445	0.598	**0.439**	0.628	5
8	1.876	2.602	2.298	1.735	**1.515**	1.748	1.629	2
9	0.852	1.447	1.014	**0.749**	0.963	1.077	0.903	3
10	0.869	1.45	0.993	**0.679**	0.888	0.928	0.893	4
11	0.952	1.519	1.045	1.063	**0.788**	0.979	0.893	2
12	0.774	1.318	0.897	0.955	**0.651**	1.006	0.758	2
13	1.393	1.439	1.347	1.409	1.239	1.398	**1.222**	1
14	1.09	1.574	1.135	1.178	**0.968**	1.215	0.985	2
15	0.738	1.081	0.775	**0.670**	0.769	0.859	0.688	2
16	0.729	1.064	0.909	1.132	**0.563**	0.867	0.679	2
17	0.717	0.939	0.752	**0.643**	0.735	1.074	0.69	2
18	1.188	1.398	1.191	**1.064**	1.255	1.256	1.135	2
Avg.	1.0075	1.5751	1.1119	1.0092	0.9322	1.0969	**0.9553**	1

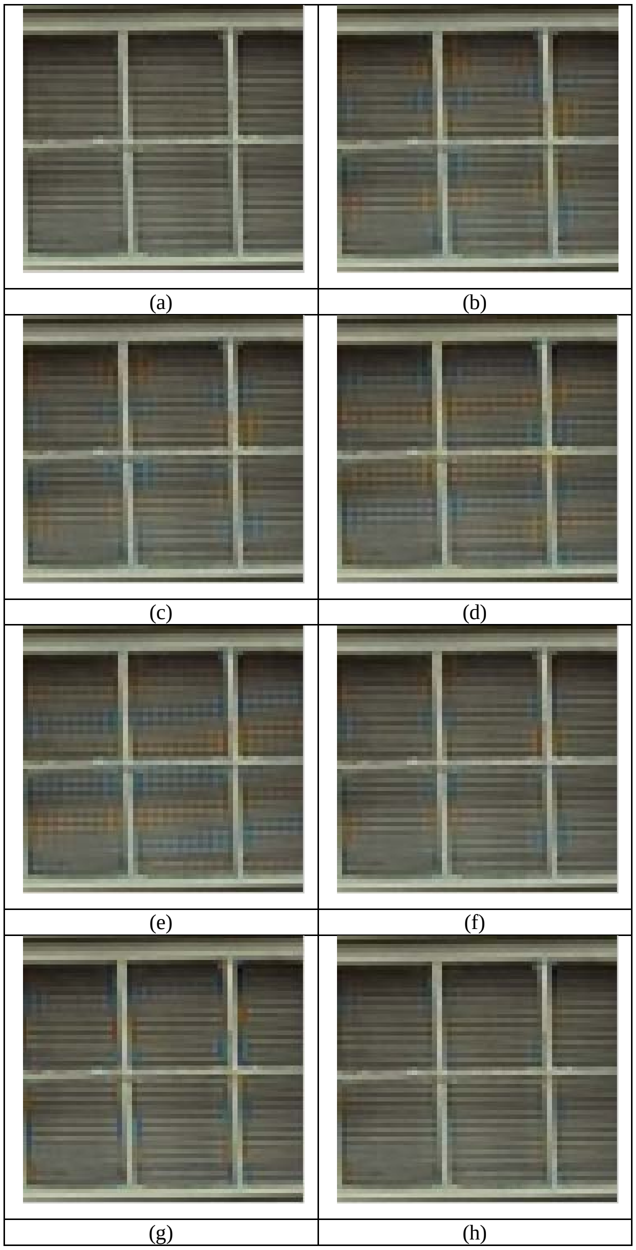


Figure 2. (a) Zoomed-in sub-image of original image #1 and the demosaicked images by:(b) DL [5]; (c) LSLCD [6]; (d) EDEP [7]; (e) AFD [8]; (f) ESF [9]; (g) LADI$_N$ and (h) LADI.

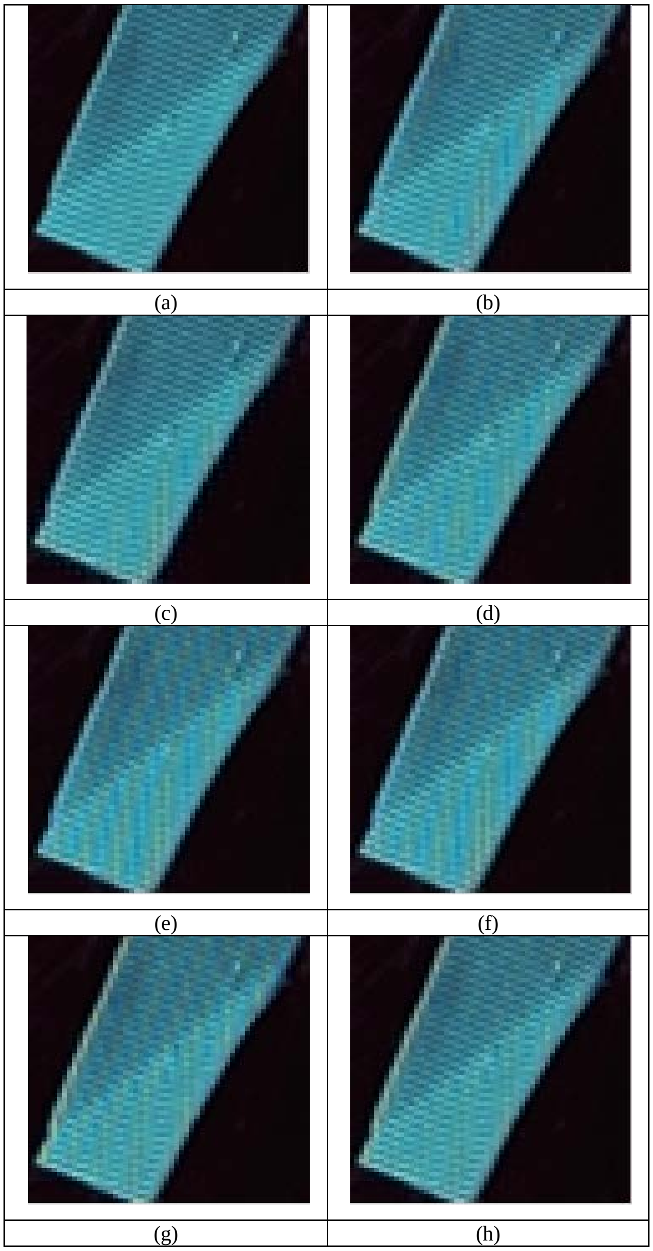


Figure 3. (a) Zoomed-in sub-image of original image #15 and the demosaicked images by:(b) DL [5]; (c) LSLCD [6]; (d) EDEP [7]; (e) AFD [8]; (f) ESF [9]; (g) LADI$_N$ and (h) LADI.