International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 6 | ISSN : 2456-3307

Color Demosaicking in Digital Image Using Nonlocal Adaptive Thresholding and Local Directional Interpolation

Kr. Nain Yadav
M.Tech Scholar, Department of Computer Science, NVPEMI, Kanpur, Uttar Pradesh, India

ABSTRACT

Single sensor digital color cameras capture only one of the three primary colors at each pixel and a process called color demosaicking (CDM) is used to reconstruct the full color images. Most CDM algorithms assume the existence of high local spectral redundancy in estimating the missing color samples. However, for images with sharp color transitions and high color saturation, such an assumption may be invalid and visually unpleasant CDM errors will occur. In this paper we exploit the image non-local redundancy to improve the local color reproduction result. First, multiple local directional estimates of a missing color sample are computed and fused according to local gradients. Then nonlocal pixels similar to the estimated pixel are searched to enhance the local estimate. An adaptive thresholding method rather than the commonly used nonlocal means filtering is proposed to improve the local estimate. This allows the final reconstruction to be performed at the structural level as opposed to the pixel level. Experimental results demonstrate that the proposed local directional interpolation and nonlocal adaptive thresholding (LDI-NAT) method outperforms many state-of-the-art CDM methods in reconstructing the edges and reducing color interpolation artifacts, leading to higher visual quality of reproduced color images.

Keywords: Color Demosaicking, Nonlocal, Sparse Representation, Image Interpolation.

I. INTRODUCTION

Single sensor (CCD/CMOS) digital color cameras capture images with a color filter array (CFA), such as the Bayer pattern CFA [1]. At each pixel, only one of the three primary colors (red, green and blue) is sampled; the missing color samples are estimated by a process called color demosaicking (CDM) to reconstruct full color images. The color reproduction quality depends on the image contents and the employed CDM algorithms [19]. Various CDM algorithms [3-18] have been proposed in the past decades. The classical second order Laplacian correction (SOLC) [3-4] algorithm is one of the benchmark CDM schemes due to its simplicity and efficiency. The recently developed methods include the successive approximation based CDM by Li [9], the adaptive homogeneity CDM by Hirakawa et al. [10], the directional linear minimum mean square-error estimation (DLMMSE) based CDM method by Zhang et al. [12], the directional filtering and a posteriori decision CDM by Menon et al. [13], the sparse representation based method by Mairal et al. [14], and the nonlocal means based self-similarity driven (SSD) method by Buades et al. [15], etc. A recent review of CDM methods can be found in [20].

Most of the existing CDM methods assume high local spectral correlations. This assumption may well be valid for images such as those in the Kodak dataset [2]. The Kodak dataset was not originally released for CDM but it has been widely used as a benchmark

dataset in evaluating CDM algorithms. Inadvertently, the Kodak dataset misled the research of CDM to some extent. It was pointed out in [15, 16, 20] that images in the Kodfak dataset have much higher spectral correlation, lower color saturation and smaller chromatic gradients than images in other datasets

In this paper, we propose to couple local directional interpolation (LDI) with nonlocal enhancement for a more effective CDM. The employed CDM strategy is very simple: initial local CDM by LDI, followed by a nonlocal enhancement process. In the initial CDM, only the local spatial-spectral correlation within a compact local window is exploited to avoid CDM errors caused by high color variations around color edges of high saturation. Since directional information is crucial for edge preservation, we use directional filters to interpolate the missing color samples. The obtained directional estimates are then fused according to the local directional gradients. The results of LDI can be augmented by exploiting non-local redundancy to reduce initial CDM errors. The similar pixels to the estimated pixel are chosen by patch matching (in practice, a relatively large local window is used), and the matched pixels are used to enhance the initial CDM result.

II. THE PROPOSED COLOR DEMOSAICKING ALGORITHM

FLOWCHART

Figure 1 illustrated the flowchart of the proposed CDM algorithm. First, an initial interpolation is applied to the green (G) channel by local direction interpolation (LDI) and fusion. Second, the nonlocal adaptive thresholding (NAT) is applied to enhance the interpolated G channel. In the 3rd step, the red (R) and blue (B) channels are initially interpolated by the help of the reconstructed G channel. Finally, NAT is applied to the R and B channels so that the whole CDM is completed.

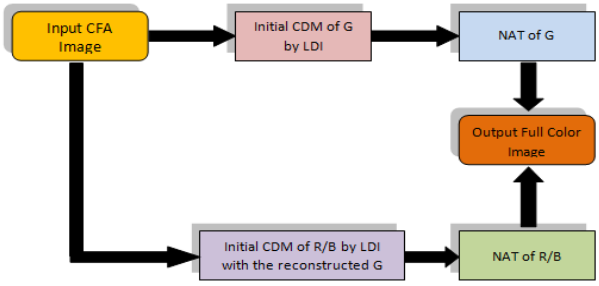
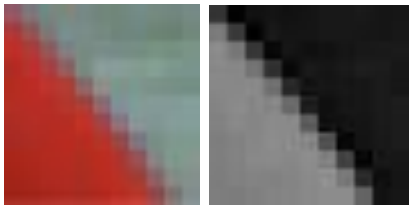
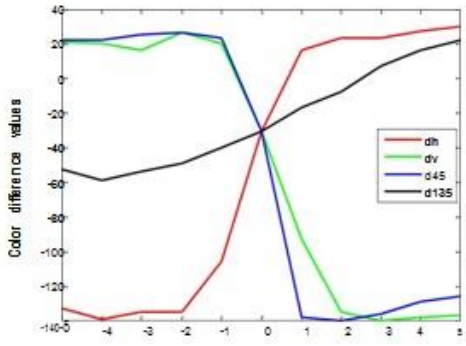


Figure 1. Flowchart of the proposed Color Demosaicking (CDM) method.

One key issue in the initial CDM is the use of local and directional information. In high saturation areas of natural images, the change of colors is abrupt. Therefore, if we use too many local neighbors to estimate the missing color samples, unexpected errors can be introduced and they can be hard to remove in the stage of nonlocal enhancement. On the other hand, the preservation of edges is crucial to the visual quality of reconstructed color images. Since edges usually have one or more dominant directions, the interpolation should be along, instead of across, the edge main directions.



(a) (b)



(c)

Figure 2. (a) A cropped and zoomed full color patch; (b) the green and red color difference image of (a);(c) the color difference signals along horizontal (dh), vertical (dv), 45⁰ diagonal (d45), and 135⁰ diagonal

(d135) directions at the center of color difference image (b).

With the above considerations, we propose an LDI scheme for initial CDM (the detailed description of LDI is in Section 2.2). Let's use an example to explain why the strategy of LDI is adopted for initial CDM. Figure 2a shows a small patch where there are sharp color transitions (from red to white) in it. Figure 2b shows the green and red color difference image (i.e. G-R) of Figure 2a. In Figure 2c, we plot the color difference signals (with the origin being the center of the patch) along four directions: horizontal (dh), vertical (dv), 45^0 diagonal (d45) and 135^0 diagonal (d135). Some observations can be made from this example.

First, the assumption of smooth color difference used in many CDM methods is invalid. Particularly, from Figure 2c we see that the color differences outside the two-pixel-wide neighborhood are very different from the center one. Therefore, using a big local window (e.g., bigger than 5×5) to estimate the missing color samples can result in unexpected errors. Second, the color edge direction information is very useful for color interpolation. From Figure 2c we see that the color difference along the 135^0 diagonal direction is much smoother than other directions, and hence it should contribute more to the color estimation. Due to the color down-sampling in the mosaic CFA pattern, the color difference signal G-R along diagonal directions cannot be directly calculated. In practice, they are estimated as the weighted average of color differences in other directions.

LOCAL DIRECTIONAL INTERPOLATION OF GREEN CHANNEL

In various CFA patterns, such as the Bayer pattern [1], the sampling frequency of G is higher than that of R and B channels. Therefore, the G channel preserves much more image structural information than the other two color channels. Usually, a better reconstruction of G will lead to a better reconstruction of R and B. As shown in Fig. 1, we will initially interpolate the G channel by using local redundancy, and then enhance it by using nonlocal redundancy.

The well-known SOLC algorithm [3, 4] is actually a directional interpolation method. In SOLC, at each R or B position two filtering outputs of G are computed along horizontal and vertical directions respectively, and then one of them is selected based on the gradients in the two directions. However, SOLC has two problems. First, it considers only two directions in the interpolation. This limits its capability in preserving edge structures along other directions. Second, SOLC simply selects one of the two directions for interpolation, but this will lose much useful information in the local area, resulting in many interpolation errors. In this section, we propose to fuse the directional information for more robust color interpolation.

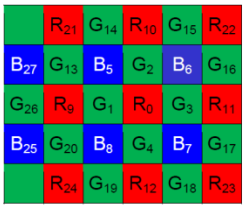


Figure 3. A CFA block.

Since there can be sharp color transitions in highly saturated regions, we use a compact local window for the initial interpolation. Refer to Fig. 3, considering a CFA block and let's focus on the red pixel R_0, where the green color is to be estimated. (The missing green colors on blue pixels can be similarly interpolated.) Intuitively, if we could know the color difference between G and R at position R_0, denoted by $d_{gr} = G_0 - R_0$, the missing green sample can then be recovered as $G_0 = R_0 + d_{gr}$. Therefore, how to estimate the color difference d_{gr} is a key in the interpolation of G.

We compute the color difference along four directions: north (n), south (s), west (w) and east (e).

Refer to Fig. 3, the four directional estimates, d_{gr}^{n}, d_{gr}^{s}, d_{gr}^{w} and d_{gr}^{e}, are calculated as follows:

$$\begin{cases} d_{gr}^{n} = G_2 - (R_0 + R_{10})/2 \\ d_{gr}^{s} = G_4 - (R_0 + R_{12})/2 \\ d_{gr}^{w} = G_1 - (R_0 + R_9)/2 \\ d_{gr}^{e} = G_3 - (R_0 + R_{11})/2 \end{cases}$$

The interpolation error of the four directional estimates relates to the edge direction and color transition at R_0. In order to evaluate which estimate is better, we calculate the gradients at R_0 along the four directions. There are many forms to define the directional gradients at R_0. We have the following considerations. First, the gradient should be calculated using the pixels from the same channel; second, to make the calculation of gradients more stable, we could involve neighboring columns/rows of the central column/row in calculation; third, the central column/row should have higher contribution to the gradient than the neighboring columns/rows. Based on the above three considerations, we use the following formula to calculate the gradients along north, south, west and east directions:

NONLOCAL ENHANCEMENT OF G CHANNEL

By using the method described in Section 2.2, an initial estimate of each missing green sample can be obtained. Since only the local redundancy in a compact local window is exploited, the interpolation may not be accurate, especially around object boundaries where sharp color or intensity changes will occur. Fortunately, in natural images there are many similar patterns or structures, while a similar structure to the given one may appear far from it. Such nonlocal redundancy can be exploited to enhance the CDM results. The nonlocal means (NLM) technique has been extensively studied and effectively used in image/video denoising and restoration [21-26], and recently it has also been successfully used in CDM [15]. In this section, we use the nonlocal redundancy to reduce the initial interpolation errors and enhance the color reproduction quality of G channel.

1. Nonlocal enhancement by NLM filtering

One straightforward way for the nonlocal enhancement of G channel is to apply NLM filtering to the interpolated green sample $\hat{G}_o$, as in many NLM based denoising works [21-25]. To this end, we search for similar pixels (can be either original green samples or interpolated green samples) to the in the recovered G image. The searching can be performed in the whole image; however, given G_0 this is computationally prohibitive and is not necessary. In practice, we search for similar pixels to $\hat{G}_o$ in a large enough window (e.g. a 31×31 window), denoted by , centered on it. The patch based G_0 method can be used to determine the similarity between $\hat{G}_o$ and other pixels in . Denote by $\mathbf{P^0}$ the s×s patch centered on $\hat{G}_o$, and by $\mathbf{P^i}$ the s×s patch centered on a green pixel G^i in . The l^1-norm distance between $\mathbf{P_0}$ and $\mathbf{P_i}$ is computed as

$$d_i = \left\| \mathbf{P}_0 - \mathbf{P}_i \right\|_1 = \frac{1}{s^2} \sum_{k=1}^{s} \sum_{l=1}^{s} \left| \mathbf{P}_0(k,l) - \mathbf{P}_i(k,l) \right| \qquad (2\text{-}7)$$

For the convenience of expression, we denote by z0 the given pixel $\hat{G}_o$ a , by zn, n=1,…,N-1, the searched similar pixels to $\hat{G}_o$, and by d^n the associated distance of z^n. The nonlocal enhancement by NLM filtering, denoted by $\hat{x}_0$, is computed as the weighted average of z_n:

output of G_0

$$\hat{x}_0 = \sum_{n=0}^{N-1} w_n z_n \qquad (2\text{-}8)$$

where the weights w_n are set as

$$w_n = \exp(-d_n/\sigma)/C \qquad (2\text{-}9)$$

with $C = \sum_{n=0}^{N-1} \exp(-d_n/\sigma)$ being the normalization factor to make the sum of w_n be 1. In Eq. (2-9), parameter σ controls the decay rate of weight w_n w.r.t. distance d_n. In the literature of image denoising, σ is usually preset according to the standard deviation of the noise in the image. In the SSD algorithm for

CDM [15], a coarse-to-fine strategy was used. The nonlocal average process is iterated three times, and the parameter σ is set smaller and smaller in the three iterations.

2 Nonlocal enhancements by NAT

The NLM filtering based nonlocal enhancement of $\hat{G}_o$ is actually the weighted average of samples z_0, z_1, …, z_{N-1}. Although it can suppress many interpolation errors generated in the initial CDM and lead to much better color reproduction than many existing CDM algorithms (refer to Section 3.2 please), it may also smooth the edges and some bad color artifacts around object boundaries can still survive.

Nonetheless, in NLM the local neighboring pixels to $\hat{G}_o$ in the patch P0, which all together form the local pattern (i.e. structure) on $\hat{G}_o$, are only used to determine the weights w^n for averaging. Actually $\mathbf{P^0}$ and the similar patches $\mathbf{P^i}$ to it also specify the variations of the local pattern on $\hat{G}_o$. This $\hat{G}_o$ information is not efficiently exploited in NLM weighting. To more effectively exploit the nonlocal redundancy, we propose a nonlocal adaptive thresholding (NAT) scheme in this section.

3. Initial Interpolation of R and B Channels

With the non-locally enhanced G channel, we first compute the initial estimates of R and B channels by exploiting the local spatial-spectral correlation, and then enhance them by nonlocal redundancy. Since the interpolations of R and B channels are symmetric, in the following we only discuss the reconstruction of B.

We interpolate the missing B samples by using a two-step strategy. First we interpolate the B samples at the R positions, and then with these interpolated B samples, all the other B samples at the G positions can be interpolated. Refer to Fig. 3, suppose we are to interpolate the missing sample B_0 at R_0. Note that all the G samples have been recovered and are available now, and we can estimate the color differences between B and G along the four diagonal directions at R_0 as:

$$\begin{cases} d_{bg}^{nw} = B_5 - G_5 \\ d_{bg}^{ne} = B_6 - G_6 \\ d_{bg}^{se} = B_7 - G_7 \\ d_{bg}^{sw} = B_8 - G_8 \end{cases}$$

where the superscripts “nw”, “ne”, “se” and “sw” represent the north-western, north-east, south-east and south-western directions, respectively.

The four directional estimates are weighted for a more robust estimate. To determine the weights, the gradients along the four directions are calculated as follows:

$$\begin{cases} \nabla_{nw} = |B_5 - B_7| + |R_{21} - R_0| + |G_5 - G_0| + \varepsilon \\ \nabla_{ne} = |B_6 - B_8| + |R_{22} - R_0| + |G_6 - G_0| + \varepsilon \\ \nabla_{se} = |B_5 - B_7| + |R_{23} - R_0| + |G_7 - G_0| + \varepsilon \\ \nabla_{sw} = |B_6 - B_8| + |R_{20} - R_0| + |G_8 - G_0| + \varepsilon \end{cases} \quad (2\text{-}15)$$

where ε is a small positive number. Like in Eq. (2-3) and Eq. (2-4), the four weights are set as

$$\bar{w}_{nw} = \frac{1}{C \cdot \nabla_{nw}}, \bar{w}_{ne} = \frac{1}{C \cdot \nabla_{ne}}, \bar{w}_{se} = \frac{1}{C \cdot \nabla_{se}}, \bar{w}_{sw} = \frac{1}{C \cdot \nabla_{sw}} \quad (2\text{-}16)$$

Where $C = \frac{1}{\nabla_{nw}} + \frac{1}{\nabla_{ne}} + \frac{1}{\nabla_{se}} + \frac{1}{\nabla_{sw}}$. Then the final blue and green color difference at position R_0 is estimated by $\hat{d}_{bg} = \bar{w}_{nw} d_{bg}^{nw} + \bar{w}_{ne} d_{bg}^{ne} + \bar{w}_{se} d_{bg}^{se} + \bar{w}_{sw} d_{bg}^{sw}$, and the missing blue component at R_0 is estimated as $\hat{B}_0 = G_0 + \hat{d}_{bg}$.

4 Nonlocal Enhancements of R and B Channels

Once the R and B channels are interpolated with the help of nonlocally enhanced G channel, they can then be enhanced by exploiting nonlocal redundancies in R and B channels respectively. The process is the same as that for the G channel. For each interpolated red (blue) sample $\hat{R}_o$(B_0), we search for similar pixels to it in a large window centered on it. The N most similar pixels to $\hat{R}_o$(B_0),

including itself, are used to enhance it via NLM or NAT.

III. EXPERIMENTAL RESULTS

1. THE MCMASTER DATASET

The Kodak image dataset [2] is widely used as a standard dataset in CDM and many other color image processing fields. The Kodak dataset contains 24 full color images, whose spatial size is 768×512. It is said that these images were originally captured by film and then digitized by scanner. However, in recent years it has been noticed that the statistics of Kodak images are very different from other natural images [15, 16, 20], e.g., the images in the McMaster dataset to be introduced. The images in Kodak dataset look smooth and less saturated, which makes them less representative for the digital color images captured by the current digital cameras and hence less representative for applications such as CDM. Specifically, the Kodak images have very high spectral correlation, are smooth in chromatic gradient and have low saturation (refer to Table 1 please). It is doubted that these images were post-processed, and they are not suitable for evaluating CDM algorithms.

Table 1. Statistics of the Kodak and the McMaster datasets.

Datasets		Kodak	McMaster
Mean Spectral Correlation	G and R	0.8712	0.7445
	G and B	0.9050	0.7114
Mean Saturation		15.6	45.81
Mean Chromatic Gradient		1.78	4.54

In this study, we use a new color image dataset, namely the McMaster dataset, for the evaluation of CDM algorithms. This dataset was established in a project of developing new CDM methods by McMaster University, Canada, in collaboration with some industry partners. It has 8 high resolution (size: 2310×1814) color images that were originally captured by Kodak film and then digitized. The scenes of the 8 images are shown in Fig. 4. Since these images have a big size, we crop 18 sub-images (size: 500×500) from them to evaluate the CDM methods. Fig. 5 shows the cropped 18 sub-images. In Table 1 we compare the mean spectral correlation, mean chromatic gradient and mean saturation[2] of the images in the two datasets. We see that the spectral correlation of the Kodak images is obviously higher than that of the McMaster dataset. The McMaster images are more saturated and there are many sharp structures with abrupt color transitions in them. Many CDM methods use the Kodak dataset as the target images in algorithm development and testing, and they assume that the color differences change smoothly. Though this assumption holds well for the Kodak dataset, we can seefrom Table 1 that it may not hold for the images in the McMaster and the source code of the proposed LDI-NLM and LDI-NAT dataset. The cropped 18 sub-images algorithms can be downloaded at

Figure 4. Scenes of the eight test images in McMaster dataset.

Figure 5. Cropped McMaster sub-images (500×500) used in the experiments. From top to bottom and left

to right, these sub-images are labeled as 1 to 18.

3.2. CDM RESULTS

In our implementation of LDI-NLM, 25 similar patches to the given patch (patch size: 5×5) are searched in a 31×31 local window. (Please note that based on our experiments, using more similar patches in NLM filtering will not improve the final CDM performance.) The parameter σ (refer to Eq. (2-9)) in the NLM filtering is set as 2.5. In our implementation of LDI-NAT, 100 similar patches to the given patch (patch size: 5×5) are searched in a 31×31 window. The threshold used in Eq. (2-13) is set as $t=0.03\times g_Y$, where g_Y is the average gradient magnitude of the similar patches.

It is well-known that PSNR is not a good indicator of CDM quality because the CDM errors mainly occur around the (color) edges, which account only a small portion of the image pixels. In [15], the Zipper Effect Ratio (ZER) was used to evaluate the color edge preservation performance of CDM. Although this metric cannot perfectly reflect the CDM quality, it works better than PSNR in evaluating the CDM performance. Table 4 shows the ZER measures of the seven competing methods. Fig. 7 presents graphically the average ZER results by various methods on the McMaster dataset. We see that LDI-NLM, SSD and LDI-NAT achieve much lower ZER values than other methods. Although SOLC and DLMMSE have similar PSNR results to SSD, their ZER measures are much worse than SSD. This also validates that PSNR is not a good metric to measure image edge preservation. Note that LDI-NLM has lower ZER values than LDI-NAT. However, LDI-NAT actually has much better edge preservation than LDI-NLM. This is because LDI-NLM results in smooth color edges, while the ZER metric favors smooth images. Nonetheless, how to define a good CDM quality metric is a very difficult problem and this is beyond the discussion of this paper.

IV. CONCLUSION

The proposed LDI-NAT algorithm was tested on the McMaster dataset in comparison with state-of-the-art CDM methods. The experimental results showed that LDI-NAT leads to visually much better demosaicked images, reducing significantly the unpleasing zipper effects and false colors that often appear in highly saturated areas.

V. REFERENCES

[1]. B. E. Bayer and Eastman Kodak Company, "Color Imaging Array," US patent 3 971 065, 1975.

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

[3]. J. E. Adams, "Intersections between color plane interpolation and other image processing functions in electronic photography," Proceedings of SPIE, vol. 2416, pp. 144-151, 1995.

[4]. J. E. Adams and J. F. Hamilton Jr., "Adaptive color plane interpolation in single color electronic camera," U. S. Patent, 5 506 619, 1996.

[5]. R. Kimmel, "Demosaicing: Image reconstruction from CCD samples," IEEE Trans. on Image Processing, vol. 8, no. 9, pp. 1221-1228, Sep. 1999.

[6]. B. K. Gunturk, Y. Altunbasak and R. M. Mersereau, "Color plane interpolation using alternating projections," IEEE Trans. on Image Processing, vol. 11, no. 9, pp. 997-1013, Sep. 2002.

[7]. R. Lukac, K. Martin, and K.N. Plataniotis, "Demosaicked image postprocessing using local color ratios," IEEE Trans. on Circuits and Syst. for Video Tech., vol. 14, no. 6, pp. 914-920, June 2004.

[8]. D. D. Muresan and T. W. Parks, "Demosaicing using optimal recovery," IEEE Trans. on Image Processing, vol. 14, no. 2, pp. 267 – 278, Feb. 2005.

[9]. Xin Li, "Demosaicing by successive approximation," IEEE Trans. on Image Processing, vol. 14, no. 3, pp. 370-379, Mar. 2005.

[10]. K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm", IEEE Trans. on Image Processing, vol. 14, no. 3, pp. 360-369, Mar. 2005.

[11]. D. Alleysson, S. Susstrunk and J. Herault, "Linear demosaicing inspired by the human visual system," IEEE Trans. on Image Processing, vol. 14, no. 4, pp. 439-449, April 2005.

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

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

[14]. J. Mairal, M. Elad, and G. Sapiro, "Sparse Representation for Color Image Restoration," IEEE Transactions on Image Processing, vol. 17, no. 1, pp. 53–69, 2009.

[15]. 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.

[16]. F. Zhang, X. Wu, X. Yang, W. Zhang and L. Zhang "Robust color demosaicking with adaptation to varying spectral correlations", IEEE Trans. on Image Processing, vol. 18, no. 12, pp. 2706-2717, Dec. 2009.

[17]. X. Wu and L. Zhang, "Color video demosaicking via motion estimation and data fusion," IEEE Trans. on Circuits and Systems for Video Technology, vol. 16, pp. 231-240, Feb. 2006.

[18]. X. Wu and L. Zhang, "Improvement of color video demosaicking in temporal domain," IEEE Trans. on Image Processing, vol. 15, Oct. 2006.

[19]. P. Longère, Xuemei Zhang, P. B. Delahunt and Davaid H. Brainard, "Perceptual assessment of demosaicing algorithm performance," Proc. of IEEE, vol. 90, no. 1, pp. 123-132, Jan. 2002.

[20]. Xin Li, B. Gunturk, L. Zhang, "Image demosaicking: a systematic survey," Visual Communications and Image Processing 2008, Proceedings of the SPIE, Volume 6822, pp. 68221J-68221J-15 (2008). San Jose, CA, USA.

[21]. A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," Multisc. Model. Simulat., vol. 4, no. 2, pp. 490.530, 2005.

[22]. S. Kindermann, S. Osher, and P. W. Jones, "Deblurring and denoising of images by nonlocal functionals," Multiscale Modeling and Simulation, vol. 4, no. 4, pp. 1091-1115, 2005.

[23]. K.Dabov, A. Foi, V.Katkovnik, K. Egiazarian, "Image denoising by sparse 3-D transform domain collaborative filtering," IEEE Transactions on Image Processing, vol. 16, no. 8, pp. 2080-2094, Aug. 2007.

[24]. T. Brox, O. Kleinschmidt, D. Cremers, "Efficient nonlocal means for denoising of textural patterns," IEEE Transactions on Image Processing, vol. 17, no. 7, pp. 1083-1092, July, 2008.

[25]. A. Buades, B. Coll, J.M Morel, "Nonlocal image and movie denoising," International Journal of Computer Vision, vol. 76, no. 2, pp. 123-139, 2008.

[26]. Xin Li and Yunfei Zheng, "Patch-based video processing: a variational Bayesian approach," IEEE Trans. on Cir. Sys. for Video Tech., vol. 19, no. 1, pp. 27-40, Jan. 2009.

Table 3. PSNR (dB) results by different CDM methods on the McMaster dataset.

Methods	SOLC [3]	AHD [10]	SA [9]	DLMMSE [12]	SSD [15]	LDI-NLM	LDI-NAT	
	R	28.26	26.02	23.53	26.94	27.28	28.81	**29.29**
	G	31.22	29.82	25.17	30.63	30.68	32.31	**32.67**
1	B	26.34	24.04	22.05	24.82	25.12	26.47	**26.71**
	R	33.68	32.47	31.63	33.30	33.61	34.66	**35.02**
	G	37.62	37.20	34.00	37.66	37.81	39.01	**39.08**
2	B	32.11	31.26	30.74	31.86	32.01	32.79	**32.92**
	R	30.64	31.10	31.47	32.60	32.81	**33.41**	33.05
	G	33.73	33.49	32.75	35.28	35.05	35.50	**35.51**
3	B	28.60	29.67	29.80	30.70	30.93	**30.99**	30.31
	R	32.80	33.76	34.59	34.70	36.36	**37.41**	36.25
	G	37.16	35.66	34.05	36.99	38.98	39.01	**40.33**
4	B	30.89	31.48	32.19	32.07	33.49	**34.02**	33.30
	R	33.61	29.52	28.60	30.38	31.10	34.50	**35.05**
	G	36.28	34.73	30.97	35.11	35.43	37.67	**38.15**
5	B	30.47	28.78	28.08	29.41	29.48	31.02	**31.16**
	R	37.14	33.92	32.23	34.98	36.09	38.59	**39.40**
	G	40.30	37.72	32.50	38.61	38.85	41.70	**43.42**
6	B	34.00	29.96	29.14	31.15	31.72	34.21	**34.97**
Average	G	38.11	37.10	34.63	38.10	38.08	39.46	**39.79**
	B	33.41	32.30	31.87	33.15	33.47	34.33	**34.38**

Table 4. Zipper Effect Ratio (ZER) by different CDM methods on the McMaster dataset.

Methods	SOLC [3]	AHD [10]	SA [9]	DLMMSE [12]	SSD [15]	LDI-NLM	LDI-NAT
1	0.2059	0.1678	0.4348	0.2021	0.0996	0.0748	0.1082
2	0.0939	0.1225	0.1673	0.1249	0.0753	0.0486	0.0682
3	0.1659	0.2336	0.4357	0.2179	0.1044	0.0815	0.1044
4	0.3475	0.3952	0.7680	0.5287	0.0915	0.0930	0.1468
5	0.0996	0.1130	0.1831	0.1144	0.0629	0.0477	0.0591
6	0.0987	0.1206	0.2121	0.1306	0.0563	0.0477	0.0477
average	0.1508	0.1949	0.3107	0.2001	0.0883	0.0633	0.0891

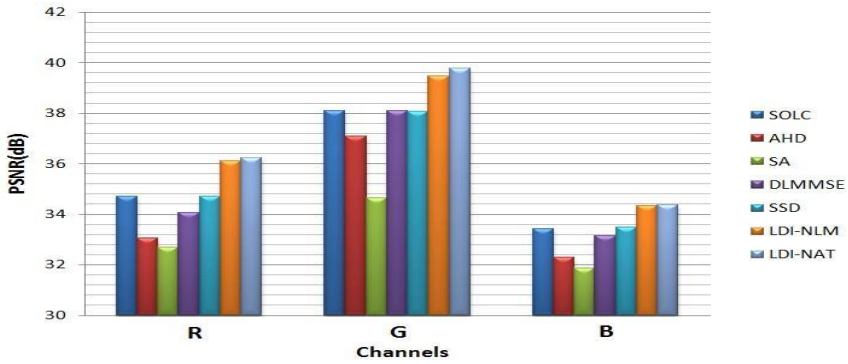


Figure 6. Graphical presentation of the average PSNR by different methods on the McMaster dataset.

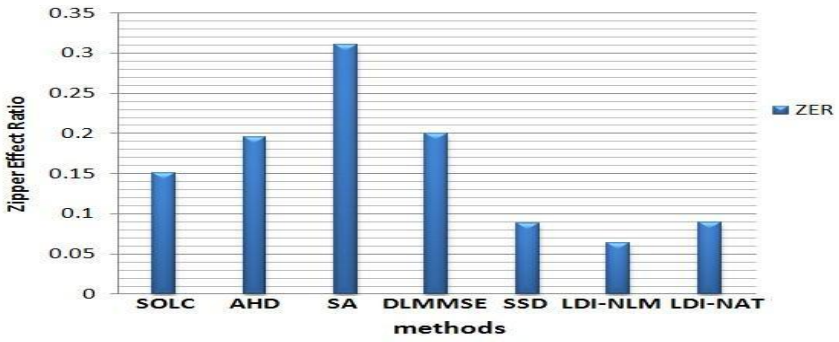


Figure 7. Graphical presentation of the average ZER by different methods on the McMaster dataset.

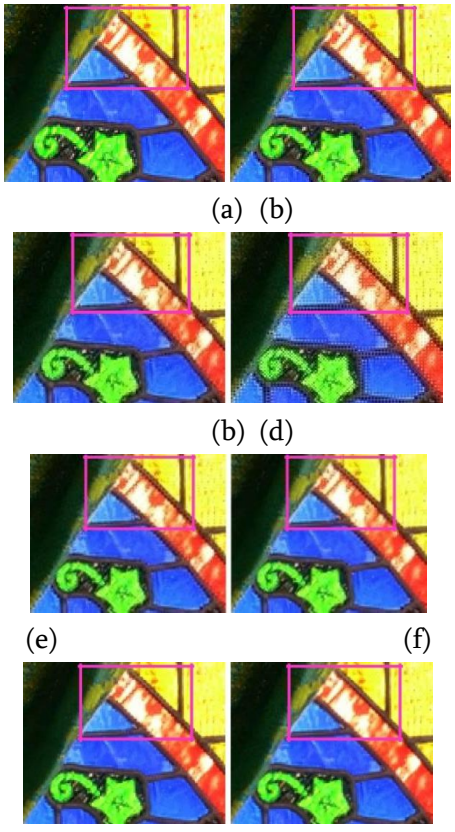


Figure 8. (a) Original image *1* and demosaicked images by (b) SOLC [3]; (c) AHD [10]; (d) SA [9]; (e) DLMMSE [12]; (f) SSD [15]; (g) the proposed LDI-NLM and (h) LDI-NAT.