

Color Demosaicking by Local Directional Interpolation and Nonlocal Adaptive Thresholding

Lei Zhang^{1,a}, Xiaolin Wu^b, Antoni Buades^c, and Xin Li^d

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.

¹ Corresponding author. Email: cszhang@comp.polyu.edu.hk. This work is supported by the Hong Kong RGC General Research Fund (PolyU 5375/09E).

^a L. Zhang is with the Dept. of Computing, The Hong Kong Polytechnic University, Hong Kong, China.

^b X. Wu is with the Dept. of Electrical and Computer Engineering, McMaster University, Canada.

^c A. Buades is with the MAP5, Université Paris Descartes, Paris, France.

^d X. Li is with the Lane Dept. of Computer Science and Electrical Engineering, West Virginia University, US.

1. 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 Hiraikawa *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 Kodak dataset have much higher spectral correlation, lower color saturation and smaller chromatic gradients than images in other datasets, e.g., the McMaster dataset used in this paper (refer to Section 3.1 for more information). Compared with the digital color images captured by current digital cameras, the images in Kodak dataset are smoother and less saturated, and hence they are less representative for the applications such as CDM. On the McMaster dataset, the simple SOLC method outperforms many lately developed more complex methods. The reason appears to be that these methods were developed aiming to reproduce the problematic Kodak images, without considering a wider range of test images.

In natural images the spectral correlation is often weak around object boundaries. Consequently, many CDM algorithms derived under the assumption of high spectral correlation may fail in areas of edges. One way to improve color reproduction near edges is to exploit the nonlocal spatial and spectral redundancies. In natural images, there can be many similar structures/patterns throughout the scene. The most similar pixels to the given one can be far from it. Thus we can relax the constraint of local neighborhood to nonlocal neighborhood when enhancing the given pixel. The nonlocal means (NLM) filters have been widely used in image processing, such as denoising and deblurring [21-26]. The mathematical framework of NLM denoising was well established by Buades *et al.* in [21], where the given pixel x is estimated as the weighted average of all pixels whose Gaussian neighborhoods look like the neighborhood of x . The recently developed SSD algorithm by Buades *et al.* [15] is an NLM based CDM scheme. In Section 3 we will see that SSD has similar PSNR results to the classical SOLC algorithm on the McMaster dataset but it achieves much better perceptual quality.

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.

A straightforward way to utilize nonlocal redundancy is NLM filtering, as in NLM denoising [21-25] and the SSD method [15]. With NLM, an initially demosaicked pixel is re-estimated as the weighted average of the similar pixels to it. Although NLM can remove much the CDM noise (i.e. initial CDM errors), it blurs sharp edges and fails to remove bad color artifacts accompanying high saturation object boundaries. To overcome these drawbacks, we propose a novel adaptive

thresholding method to make better use of non-local redundancy than the NLM filtering. Different from NLM filtering, which applies weighted average directly to the pixel to be enhanced, we model the local patch centered on the pixel as a signal vector and compute the statistics of this vector for processing. By using the nonlocal redundancy, we adaptively compute the optimal transformation domain in which the given patch is de-correlated, and then apply soft thresholding in the transformation domain for filtering. The experimental results in Section 3 clearly demonstrate that the proposed LDI and nonlocal adaptive thresholding (NAT) based method outperforms most of the existing CDM methods, including the recently developed NLM based SSD algorithm. Compared with NLM filtering, NAT works on the structural level instead of the pixel level. Therefore, it preserves sharp edges much better and removes more color artifacts than NLM.

The rest of the paper is organized as follows. Section 2 describes in detail the proposed LDI-NAT scheme for CDM. Section 3 presents the experimental results. Section 4 concludes the paper.

2. The Proposed Color Demosaicking Algorithm

2.1. Strategy and Flowchart

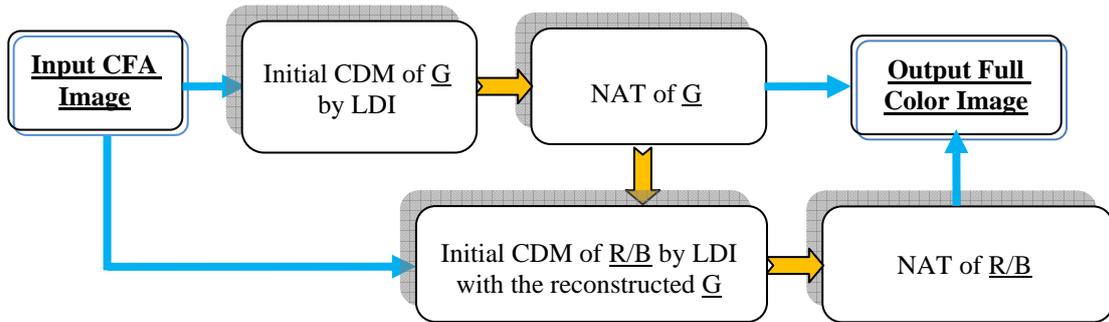


Figure 1: Flowchart of the proposed color demosaicking (CDM) method.

Fig. 1 illustrates the flowchart of the proposed CDM algorithm. First, an initial interpolation is applied to the green (G) channel by local directional interpolation (LDI) and fusion. Second, the nonlocal adaptive thresholding (NAT) is applied to enhance the interpolated G channel. In the third 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.

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.

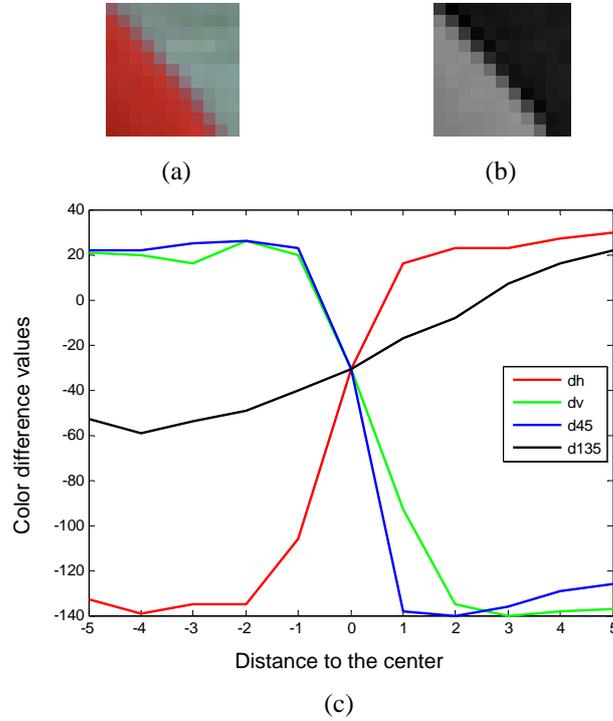


Figure 2: (a) A cropped and zoomed full color patch; (b) the green and red color difference image of (a); (c) the color difference signals along horizontal (dh), vertical (dv), 45° diagonal (d45), and 135° diagonal (d135) directions at the center of color difference image (b).

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° diagonal (d45) and 135° 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. In other words, a compact local window should be used in the initial CDM of high saturation areas. Second, the color edge direction information is very useful for color interpolation. From Figure 2c we see that the color difference along the 135° 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.

2.2. 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.

	R ₂₁	G ₁₄	R ₁₀	G ₁₅	R ₂₂
B ₂₇	G ₁₃	B ₅	G ₂	B ₆	G ₁₆
G ₂₆	R ₉	G ₁	R ₀	G ₃	R ₁₁
B ₂₅	G ₂₀	B ₈	G ₄	B ₇	G ₁₇
	R ₂₄	G ₁₉	R ₁₂	G ₁₈	R ₂₃

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:

$$\begin{cases} d_{gr}^n = G_2 - (R_0 + R_{10})/2 \\ d_{gr}^s = G_4 - (R_0 + R_{12})/2 \\ d_{gr}^w = G_1 - (R_0 + R_9)/2 \\ d_{gr}^e = G_3 - (R_0 + R_{11})/2 \end{cases} \quad (2-1)$$

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:

$$\begin{cases} \nabla_n = |G_2 - G_4| + |R_0 - R_{10}| + \frac{1}{2}|G_1 - G_{14}| + \frac{1}{2}|G_3 - G_{15}| + \varepsilon \\ \nabla_s = |G_2 - G_4| + |R_0 - R_{12}| + \frac{1}{2}|G_1 - G_{19}| + \frac{1}{2}|G_3 - G_{18}| + \varepsilon \\ \nabla_w = |G_1 - G_3| + |R_0 - R_9| + \frac{1}{2}|G_2 - G_{13}| + \frac{1}{2}|G_4 - G_{20}| + \varepsilon \\ \nabla_e = |G_1 - G_3| + |R_0 - R_{11}| + \frac{1}{2}|G_2 - G_{16}| + \frac{1}{2}|G_4 - G_{17}| + \varepsilon \end{cases} \quad (2-2)$$

where ε is a small positive number to avoid the gradient being zero.

In general, a bigger gradient along a direction means more variations in that direction and hence it is more difficult to accurately estimate the color difference, vice versa. Therefore, we can use the gradients as indices to weight the four estimates into a final estimate. An optimal weighting scheme needs to know the joint distribution of the gradient and the color difference. However, such information is unknown in advance or hard to estimate online. In this paper, we simply let the weight assigned to a directional estimate be inversely proportional to the gradient along that direction:

$$w_n = \frac{1}{\nabla_n}, w_s = \frac{1}{\nabla_s}, w_w = \frac{1}{\nabla_w}, w_e = \frac{1}{\nabla_e} \quad (2-3)$$

We then normalize the four weights to make the sum of them be 1. There is

$$\bar{w}_n = \frac{w_n}{C}, \bar{w}_s = \frac{w_s}{C}, \bar{w}_w = \frac{w_w}{C}, \bar{w}_e = \frac{w_e}{C} \quad (2-4)$$

where $C = w_n + w_s + w_w + w_e$. The four directional estimates are then fused into one estimation:

$$\hat{d}_{gr} = \bar{w}_n d_{gr}^n + \bar{w}_s d_{gr}^s + \bar{w}_w d_{gr}^w + \bar{w}_e d_{gr}^e \quad (2-5)$$

Finally, the missing green component at R_0 can be estimated as

$$\hat{G}_0 = R_0 + \hat{d}_{gr} \quad (2-6)$$

By applying the above procedures to all the R and B positions, we can reconstruct the G channel.

2.3. 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.

A. 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}_0 , 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 given \hat{G}_0 in the recovered G image. The searching can be performed in the whole image; however, this is computationally prohibitive and is not necessary. In practice, we search for similar pixels to \hat{G}_0 in a large enough window (e.g. a 31×31 window), denoted by Ω , centered on it. The patch based method can be used to determine the similarity between \hat{G}_0 and other pixels in Ω . Denote by \mathbf{P}_0 the $s \times s$ patch centered on \hat{G}_0 , and by \mathbf{P}_i the $s \times s$ patch centered on a green pixel G_i in Ω . The l_1 -norm distance between \mathbf{P}_0 and \mathbf{P}_i is computed as

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

In general, the smaller the distance d_i is, the more similar G_i is to \hat{G}_0 . Based on d_i , we select the N most similar pixels to \hat{G}_0 (including \hat{G}_0 itself) for the nonlocal enhancement of \hat{G}_0 .

For the convenience of expression, we denote by z_0 the given pixel \hat{G}_0 , by $z_n, n=1, \dots, N-1$, the searched similar pixels to \hat{G}_0 , and by d_n the associated distance of z_n . The nonlocal enhancement output of \hat{G}_0 by NLM filtering, denoted by \hat{x}_0 , is computed as the weighted average of z_n :

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

where the weights w_n are set as

$$w_n = \exp(-d_n/\sigma) / C \quad (2-9)$$

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.

B. Nonlocal enhancement by NAT

The NLM filtering based nonlocal enhancement of \hat{G}_0 is actually the weighted average of samples z_0, z_1, \dots, 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}_0 in the patch \mathbf{P}_0 , which all together form the local pattern (i.e. structure) on \hat{G}_0 , 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}_0 . This 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.

By viewing the initial CDM error as additive noise, the initial CDM output can be modeled as $\mathbf{y}=\mathbf{x}+\mathbf{v}$, where \mathbf{x} is the true signal to be restored, \mathbf{v} is additive noise, and \mathbf{y} is the initial CDM result. To robustly estimate the original signal \mathbf{x} from the degraded observation \mathbf{y} , a regularized solution is often desired such that $\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} J(\mathbf{x})$ s.t. $\|\mathbf{y} - \mathbf{x}\|_2^2 \leq \tau$, where $J(\mathbf{x})$ is the regularization term and τ is a small number. For example, in the total variational (TV) based image restoration [27-29, 31], $J(\mathbf{x})$ is the l_1 -norm of the gradients of \mathbf{x} . Recently, the sparsity prior of \mathbf{x} has been successfully used for image restoration [32-33, 37-40, 14]. By assuming that the signal \mathbf{x} can be sparsely coded (i.e.

represented) by a dictionary of atoms Ψ , i.e., $\mathbf{x} \approx \Psi \boldsymbol{\alpha}$ and most of the coefficients in $\boldsymbol{\alpha}$ are small, the sparsity based estimation of \mathbf{x} from \mathbf{y} can be obtained via l_1 -norm minimization:

$$\hat{\boldsymbol{\alpha}} = \arg \min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \quad \text{s.t.} \quad \|\mathbf{y} - \Psi \boldsymbol{\alpha}\|_2 \leq \varepsilon \quad (2-10)$$

The above l_1 -norm minimization problem can be solved by standard convex optimization techniques [41] or by the iterative shrinkage methods [40]. The sparse representation modeling has led to many interesting results in image processing, such as compressive sensing [34-36] and denoising [37, 39].

In our problem of the nonlocal enhancement of \hat{G}_0 , we denote by $\mathbf{y}_0 = [y_0, y_1, \dots, y_{M-1}]^T$, where $M = s^2$. The column vector \mathbf{y}_0 contains the samples in the $s \times s$ patch \mathbf{P}_0 centered on \hat{G}_0 , and it can be viewed as the observation of the unknown true signal $\mathbf{x}_0 = [x_0, x_1, \dots, x_{M-1}]^T$. Then we have $\mathbf{y}_0 = \mathbf{x}_0 + \mathbf{u}_0$, where \mathbf{u}_0 represents the initial CDM error. Once a good estimation of \mathbf{x}_0 , denoted by $\hat{\mathbf{x}}_0$, can be made from \mathbf{y}_0 , the nonlocal enhancement of \hat{G}_0 , denoted by $\hat{\mathbf{x}}_0$, can be readily extracted from $\hat{\mathbf{x}}_0$. Since the elements in patch \mathbf{P}_0 are highly correlated, it can be assumed that signal \mathbf{x}_0 is sparse in some domain Ψ , i.e., $\mathbf{x}_0 \approx \Psi \boldsymbol{\alpha}_0$ and $\boldsymbol{\alpha}_0$ is a sparse coefficient vector. The enhancement of \hat{G}_0 can be modeled as

$$\hat{\boldsymbol{\alpha}}_0 = \arg \min_{\boldsymbol{\alpha}_0} \|\boldsymbol{\alpha}_0\|_1 \quad \text{s.t.} \quad \|\mathbf{y}_0 - \Psi \boldsymbol{\alpha}_0\|_2 \leq \tau \quad (2-11)$$

Once $\hat{\boldsymbol{\alpha}}_0$ is optimized, the estimated signal can be obtained as $\hat{\mathbf{x}}_0 = \Psi \hat{\boldsymbol{\alpha}}_0$.

Now the question is how to determine the sparse domain Ψ in Eq. (2-11) to solve $\hat{\boldsymbol{\alpha}}_0$. Although the wavelet bases or the Fourier bases are often used, these analytically designed bases cannot effectively characterize the so many different local patterns across the image. The dictionary learning [37-38] methods have been recently proposed to learn an over-complete dictionary of bases from a training dataset to span the sparse domain. Nonetheless, for a given signal \mathbf{y}_0 , many atoms in the learned over-complete dictionary will be irrelevant, while the l_1 -norm minimization needs much computational cost. With these considerations, in this paper we propose the NAT scheme to solve Eq. (2-11) with nonlocal redundancy.

Recall that after the nonlocal similar pixels searching to \hat{G}_0 , we obtain $N-1$ similar patches to \mathbf{P}_0 . The vector \mathbf{y}_0 is formed by stretching \mathbf{P}_0 , and similarly we can form another $N-1$ column vectors by stretching \mathbf{P}_i , $i=1,2,\dots,N-1$. Denote by $\mathbf{y}_0=[y_{0,0}, y_{0,1}, \dots, y_{0,M-1}]^T$ the vector formed by \mathbf{P}_0 , and by $\mathbf{y}_i=[y_{i,0}, y_{i,1}, \dots, y_{i,M-1}]^T$ the vectors formed by other patches. Then an $M \times N$ data matrix \mathbf{Y} can be established by $\mathbf{Y}=[\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{N-1}]$. Each row of \mathbf{Y} is then centralized by subtracting its mean value. For the convenience of expression, we still use symbol \mathbf{Y} in the following development.

Since $\mathbf{y}_i = \mathbf{x}_i + \mathbf{v}_i$, where \mathbf{x}_i is the unknown true signal and \mathbf{v}_i is the initial CDM error, we have $\mathbf{Y}=\mathbf{X}+\mathbf{V}$, where $\mathbf{X}=[\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{N-1}]$ and $\mathbf{V}=[\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_{N-1}]$. A good domain Ψ for \mathbf{X} should be a domain where the vectors \mathbf{x}_i could be sparsely coded; that is, $\mathbf{X} \approx \Psi \mathbf{A}$ and \mathbf{A} is a sparse matrix. Since only the observation of \mathbf{X} , i.e. \mathbf{Y} , is available, we set the objective function to determine Ψ as:

$$\arg \min_{\mathbf{A}, \Psi} \|\mathbf{A}\|_1 \quad \text{s.t.} \quad \|\mathbf{Y} - \Psi \mathbf{A}\|_F \leq \tau \quad (2-12)$$

where $\|\cdot\|_F$ is the Frobenius norm .

Eq. (2-12) is a joint optimization problem of \mathbf{A} and Ψ , which can be solved by optimizing \mathbf{A} and Ψ alternatively. Considering that the average power of CDM error \mathbf{V} is not seriously high (but the resulting color artifacts can be visually very displeasing), here we propose an efficient solution to Eq. (2-12). By using singular value decomposition (SVD), we can factorize \mathbf{Y} as $\mathbf{Y}=\Phi \mathbf{\Gamma}$, where Φ is an orthonormal matrix spanned by the eigenvectors of the covariance matrix of \mathbf{Y} (i.e. $\mathbf{Y}\mathbf{Y}^T$), and $\mathbf{\Gamma}=\Phi^T \mathbf{Y}$ is the projection of \mathbf{Y} over Φ^T . We let the desired dictionary $\Psi=\Phi$. If we also let $\mathbf{A}=\mathbf{\Gamma}$, then the constraint $\|\mathbf{Y} - \Psi \mathbf{A}\|_F = \|\mathbf{Y} - \Phi \mathbf{\Gamma}\|_F = 0 \leq \tau$ is perfectly satisfied but $\|\mathbf{A}\|_1$ will have a certain amount so that $\arg \min_{\mathbf{A}} \|\mathbf{A}\|_1$ is not optimized. Thus $\mathbf{\Gamma}$ needs to be further processed for a better solution to \mathbf{A} .

With $\mathbf{Y}=\mathbf{X}+\mathbf{V}$, we have $\mathbf{\Gamma}=\Phi^T \mathbf{Y}=\Phi^T \mathbf{X}+\Phi^T \mathbf{V}=\mathbf{\Gamma}_X+\mathbf{\Gamma}_V$, where $\mathbf{\Gamma}_X=\Phi^T \mathbf{X}$ and $\mathbf{\Gamma}_V=\Phi^T \mathbf{V}$. Φ^T will de-correlate true signal \mathbf{X} , and many coefficients in $\mathbf{\Gamma}_X$ will be small, while there are a few significant coefficients in $\mathbf{\Gamma}_X$ and they are mainly the projection coefficients of \mathbf{X} on the eigenvectors associated with the most significant eigenvalues of $\mathbf{Y}\mathbf{Y}^T$. The CDM errors \mathbf{V} are very like random noise, and thus

the energy of Γ_V will be evenly spread over the domain spanned by Φ^J . Therefore, we could apply a soft threshold t to Γ to remove Γ_V from Γ so that the desired \mathcal{A} can be obtained as follows:

$$\mathcal{A}(i, j) = \begin{cases} \text{sign}(\Gamma(i, j)) \cdot (|\Gamma(i, j)| - t) & \text{if } \Gamma(i, j) > t \\ 0 & \text{if } \Gamma(i, j) \leq t \end{cases} \quad (2-13)$$

Actually soft-thresholding is widely used to solve the l_1 -norm minimization problems [30-31, 40].

With Eq.(2-13), the term $|\mathcal{A}|_1$ is much reduced while the term $\|\mathbf{Y} - \Psi\mathcal{A}\|_F$ can be still controlled within a small range τ , and finally a good solution to Eq.(2-12) is obtained.

The selection of threshold t depends on the CDM error level in V . In practice, we can estimate t as follows. The CDM accuracy of a local area is closely related to its local smoothness. If the local area is smooth, usually the CDM error will be low, and vice versa. Therefore, we empirically estimate the CDM error level based on the local intensity variation. We calculate the average gradient magnitude of all patches in Y and denoted it by g_Y , and then we let $t = c \cdot g_Y$, where c is a constant. By experience, we set $c = 0.03$ in the experiments.

Once the solution to \mathcal{A} in Eq. (2-12) is obtained, the desired solution $\hat{\mathbf{a}}_0$ in Eq.(2-11) is obtained by extracting it from \mathcal{A} . The nonlocal enhancement result of \hat{G}_0 is $\hat{\mathbf{x}}_0 = \Psi \cdot \hat{\mathbf{a}}_0$. From the above description, we can see that NLM applies nonlocal enhancement to \hat{G}_0 by weighted averaging, while NAT applies nonlocal enhancement to the local patch centered on \hat{G}_0 . In other words, NAT lifts NLM from the pixel level to the structure level. Consequently, NAT can reconstruct much better the image edges than NLM, as we will see in the experimental results in Section 3.

2.4. 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:

$$\begin{cases} d_{bg}^{nw} = B_5 - G_5 \\ d_{bg}^{ne} = B_6 - G_6 \\ d_{bg}^{se} = B_7 - G_7 \\ d_{bg}^{sw} = B_8 - G_8 \end{cases} \quad (2-14)$$

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:

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

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

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

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 $\hat{B}_0 = G_0 + \hat{d}_{bg}$.

Once the B samples at the R positions are interpolated as described above, we can consequently interpolate the B samples at all the other G positions. Take the position G_1 in Fig. 3 as an example. Note that the blue samples at R_9 and R_0 have been interpolated, and we denote them as \hat{B}_9 and \hat{B}_0 .

The directional estimates of the blue and green color difference at G_1 are computed as

$$\begin{cases} d_{bg}^n = B_5 - G_5 \\ d_{bg}^s = B_8 - G_8 \\ d_{bg}^w = \hat{B}_9 - G_9 \\ d_{bg}^e = \hat{B}_0 - G_0 \end{cases} \quad (2-17)$$

The gradients at position G_1 along the four directions are calculated as:

$$\begin{cases} \nabla_n = |G_{14} - G_1| + |B_5 - B_8| + \frac{1}{2}|R_{21} - R_9| + \frac{1}{2}|R_{10} - R_0| + \varepsilon \\ \nabla_s = |G_{19} - G_1| + |B_5 - B_8| + \frac{1}{2}|R_{20} - R_9| + \frac{1}{2}|R_{12} - R_0| + \varepsilon \\ \nabla_w = |G_1 - G_{26}| + |R_0 - R_9| + \frac{1}{2}|B_{27} - B_5| + \frac{1}{2}|B_{25} - B_8| + \varepsilon \\ \nabla_e = |G_1 - G_3| + |R_0 - R_9| + \frac{1}{2}|B_5 - B_6| + \frac{1}{2}|B_8 - B_7| + \varepsilon \end{cases} \quad (2-18)$$

The associated four weights for the four directions are computed in the same way as in Eq. (2-3) and

Eq. (2-4), and the fused color difference is obtained as $\hat{d}_{bg} = \bar{w}_n d_{bg}^n + \bar{w}_s d_{bg}^s + \bar{w}_w d_{bg}^w + \bar{w}_e d_{bg}^e$. Finally,

the missing B sample at position G_1 is interpolated by $\hat{B}_1 = G_1 + \hat{d}_{bg}$.

2.5. Nonlocal Enhancement 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}_0 (\hat{B}_0), we search for similar pixels to it in a large window centered on it. The N most similar pixels to \hat{R}_0 (\hat{B}_0), including itself, are used to enhance it via NLM or NAT.

3. Experimental Results

3.1. 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 saturation² 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 see

² The mean saturation is computed as follows. For each pixel with color components $\{r, g, b\}$, its saturation is computed as $s = \sqrt{\frac{((r-y)^2 + (g-y)^2 + (b-y)^2)}{3}}$, where $y = (r+g+b)/3$. The mean saturation is obtained by averaging the saturation of all pixels. The mean chromatic gradient is computed as follows. We first convert the image into the YUV space. The chromatic gradient of each pixel is set as the modulus of the gradient in the U and V channels. Then the mean chromatic gradient is the obtained by averaging over the whole image.

from Table 1 that it may not hold for the images in the McMaster dataset. The cropped 18 sub-images and the source code of the proposed LDI-NLM and LDI-NAT algorithms can be downloaded at http://www.comp.polyu.edu.hk/~cslzhang/CDM_Dataset.htm.



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

We evaluate the performance of various CDM schemes on the McMaster dataset. We denote by “LDI-NLM” the proposed LDI and NLM based CDM method, and by “LDI-NAT” the LDI and NAT based CDM method. The following representative CDM methods are used for comparison: the second order Laplacian correction (SOLC) method [3-4]; the adaptive homogeneity-directed (AHD) method [10], the successive approximation (SA) method [9], the directional linear minimum mean square-error estimation (DLMMSE) method [12], and the self-similarity driven (SSD) method [15]. Considering that the SSD, LDI-NLM and LDI-NAT algorithms all exploit the nonlocal redundancy, in Table 2 we summarize the procedures of the three algorithms for a better understanding of the common points and differences between them. These nonlocal methods involve a step of similar patch searching, which is one of the main sources of computational cost. Therefore, SSD, LDI-NLM and LDI-NAT have higher complexity than the local methods SOLC, AHD, SA and DLMMSE.

Suppose that the same nonlocal similar patch searching algorithm is used for SSD, LDI-NLM and LDI-NAT, then LDI-NLM has similar complexity to SSD because both of them use weighted average to exploit the nonlocal redundancy. However, LDI-NAT has higher complexity than SSD because it uses PCA to exploit the nonlocal redundancy. The data matrix Y formed by nonlocal similar patches is of size $M \times N$, and the covariance matrix of it is of size $M \times M$. In the PCA transformation, the SVD of the covariance matrix is required and the complexity is $O(M^3)$, which is much higher than that of weighted average. Hence, the proposed LDI-NAT has the highest complexity among the competing methods, while LDI-NLM and SSD have similar complexity.

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.

Table 2: Summary of the SSD, LDI-NLM and LDI-NAT algorithms.

Methods	SSD	LDI-NLM	LDI-NAT
Procedures	<ol style="list-style-type: none"> 1. The full color image is initially interpolated by bilinear interpolator. Denote it by \mathbf{u}_0. 2. For $\sigma = \{16, 4, 1\}$ <ol style="list-style-type: none"> 2a. Apply non-local means filtering to \mathbf{u}_0 with scale parameter σ. 2b. Convert \mathbf{u}_0 into YUV color space and apply chromatic regularization to U and V channels. Transform the regularized image back to RGB color space, and denote it by \mathbf{u}. 2c. Let $\mathbf{u}_0 = \mathbf{u}$. <p>End</p>	<ol style="list-style-type: none"> 1. The G channel is initially recovered by LDI. 2. The G channel is non-locally enhanced by NLM filtering. 3. The R or B channel is initially recovered by LDI with the non-locally recovered G. 4. The R or B channel is non-locally enhanced by NLM filtering. 	<ol style="list-style-type: none"> 1. The G channel is initially recovered by LDI. 2. The G channel is non-locally enhanced by NAT with soft-thresholding. 3. The R or B channel is initially recovered by LDI with the non-locally recovered G. 4. The R or B channel is non-locally enhanced by NAT with soft-thresholding.
The way to use nonlocal information	<ol style="list-style-type: none"> 1. A coarse-to-fine strategy is used to exploit the non-local redundancy iteratively. In each iteration, the similar pixels to the given one are weighted as the updated estimation. 2. In each iteration, the RGB color space is transformed into the YUV space for chromatic regularization. 	<ol style="list-style-type: none"> 1. The R, G and B channels are enhanced separately. 2. The similar pixels to the given one are weighted, while the weights are determined based on the distances between similar patches. 	<ol style="list-style-type: none"> 1. The R, G and B channels are enhanced separately. 2. The pixels in the given patch are taken as a vector signal, which is soft-thresholded in an adaptively computed sparse domain based on the statistics of similar patches.

In the experiments, we down-sampled the original color images into CFA images according to the Bayer pattern, and then reconstructed the full color images from the CFA mosaic data by using the seven methods. Table 3 lists the PSNR results. We see that the proposed LDI-NLM and LDI-NAT algorithms achieve much higher PSNR measures than other competing algorithms in almost every channel of all the test images. Although the classical SOLC is simple, it achieves almost the same PSNR results as the recently developed SSD scheme, while SOLC and SSD outperform the other three methods in the competition. The DLMMSE has similar PSNR results to SOLC and SSD, and the AHD and SA algorithms have the lowest PSNR measures. Fig. 6 presents graphically the average PSNR results by various methods on the McMaster dataset.

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.

Figs. 8 to 11 show the cropped and zoomed CDM results of the seven methods on images 1, 5, 6 and 16. It can be clearly seen that the proposed two algorithms, especially LDI-NAT, yield much better CDM outputs than the other five methods. The methods SA, AHD and DLMMSE produce many zipper effects and false colors because they assume smooth color differences but this assumption does not hold well on the McMaster dataset. The SOLC uses a compact (five-tap) filter to interpolate the missing colors, which makes it free of many interpolation errors caused by abrupt color changes. However, SOLC neither fully exploits the local redundancy nor uses any nonlocal redundancy for CDM. The SSD exploits the nonlocal redundancy to iteratively recover the color information but it is not effective in using the image local directional information. As a result, both SOLC and SSD still produce many visible color artifacts, which can be clearly observed in Figs. 8 to 11. The proposed LDI-NLM and LDI-NAT exploit effectively the image local redundancy and edge direction information in the initial interpolation, and exploit the non-local similarity to enhance the CDM output. They reduce significantly the CDM errors and artifacts, recovering more faithfully the missing color samples than SOLC and SSD. Their higher PSNR and lower ZER measures in Tables 3 and 4 also validate their powerful capability in color reproduction.

At last, let's compare the performance of LDI-NLM and LDI-NAT. As summarized in Table 2, LDI-NLM exploits the nonlocal redundancy by NLM filtering. NLM filtering is powerful in smoothing the initial CDM noise; however, it may also smooth the edges. In addition, for strong

zippers and color artifacts caused by sharp color transition in highly saturated areas, NLM filtering is not effective to remove them. Different from NLM filtering, LDI-NAT processes the patch centered on the given pixel as a whole to better preserve the local pattern. The nonlocal redundancy is used to compute the sparse domain, and adaptive soft-thresholding is used to remove the initial CDM errors. Compared with NLM filtering, the NAT exploits the structural statistics in the similar patches, and hence it can more effectively preserve the edges and reduce the zipper effects and false colors. From Table 3, we see that for images with more smooth areas (e.g. images 8, 9, 13 and 18), the overall PSNR measures of LDI-NLM can be higher than LDI-NAT. Nonetheless, in the smooth areas both the two methods can have good visual quality. In areas with high color variations, LDI-NAT leads to much better CDM outputs. This can be more clearly seen in the images with more color edges, e.g. images 1, 5, 6, and 16.

4. Conclusion

This paper presented a powerful color demosaicking (CDM) scheme by exploiting effectively both the local spectral correlation and the non-local similarity in the color filter array (CFA) image. Many previous CDM methods assume the high local spectral redundancy in the color interpolation. Such an assumption, however, fails for images with sharp color transitions and high color saturation. Fortunately, the nonlocal redundancy can be used to compensate for the lack of local redundancy in CDM. We first computed four directional local estimates of a missing color component, and fused them into one estimate based on the local directional gradients. After the local directional interpolation (LDI), the nonlocal similar pixels to the given one were searched to enhance the initial CDM results. Apart from the commonly used nonlocal means (NLM) filtering technique, a nonlocal adaptive thresholding (NAT) scheme was proposed to better preserve the local structure while reducing the initial CDM errors. 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.

Reference

- [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, "Demosaiicing: 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, "Demosaiiced 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, "Demosaiicing using optimal recovery," *IEEE Trans. on Image Processing*, vol. 14, no. 2, pp. 267 – 278, Feb. 2005.
- [9] Xin Li, "Demosaiicing 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, "Demosaiicing 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.
- [27] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, pp. 259–268, 1992.
- [28] P. Blomgren and T. Chan, "Color TV: Total variation methods for restoration of vector-valued images," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 304–309, Mar. 1998.

- [29] T. Chan, S. Esedoglu, F. Park, and A. Yip, "Recent developments in total variation image restoration," in *Mathematical Models of ComputerVision*, N. Paragios, Y. Chen, and O. Faugeras, Eds. New York: Springer Verlag, 2005.
- [30] D.L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. on Inform. Theory*, vol. 41, no. 3, pp. 613-627, 1995.
- [31] A. Chambolle, R.A. DeVore, N.-Y. Lee and B.J. Lucier, "Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 319-335, 1998.
- [32] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129-159, 2001.
- [33] D. Donoho and M. Elad, "Optimal sparse representation in general (nonorthogonal) dictionaries via L_1 minimization", *Proc. Nat. Aca. Sci.*, vol. 100, pp. 2197-2202, Mar. 2003.
- [34] D. Donoho, "Compressed sensing," *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 1289 - 1306, April 2006.
- [35] E. Candès and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. on Information Theory*, 52(12), pp. 5406 - 5425, December 2006.
- [36] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. on Information Theory*, 52(2) pp. 489 - 509, February 2006
- [37] M. Aharon, M. Elad, and A.M. Bruckstein, "The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311-4322, November 2006.
- [38] R. Rubinstein, A.M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of IEEE, Special Issue on Applications of Compressive Sensing & Sparse Representation*, vol. 98, no. 6, pp. 1045-1057, June, 2010.
- [39] M. Protter and M. Elad, "Image sequence denoising via sparse and redundant representations," *IEEE Trans. on Image Processing*, Vol. 18, No. 1, Pages 27-36, January 2009.
- [40] J.M. Bioucas Dias, and M.A.T. Figueiredo, "A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Trans. Image Proc.*, vol.16, No.12, pp.2992-3004, 2007.
- [41] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

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	
1	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
	B	26.34	24.04	22.05	24.82	25.12	26.47	26.71
2	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
	B	32.11	31.26	30.74	31.86	32.01	32.79	32.92
3	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
	B	28.60	29.67	29.80	30.70	30.93	30.99	30.31
4	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
	B	30.89	31.48	32.19	32.07	33.49	34.02	33.30
5	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
	B	30.47	28.78	28.08	29.41	29.48	31.02	31.16
6	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
	B	34.00	29.96	29.14	31.15	31.72	34.21	34.97

7	R	33.85	35.64	37.03	38.30	36.61	36.28	36.09
	G	36.34	37.36	40.39	40.70	37.62	37.66	37.41
	B	32.45	35.07	36.22	37.29	36.38	34.59	34.49
8	R	34.87	34.15	35.31	35.45	35.31	36.89	36.31
	G	39.09	39.45	38.49	41.43	40.34	40.44	40.29
	B	35.04	35.79	35.82	36.99	36.76	36.84	36.67
9	R	34.36	31.54	30.71	32.39	33.72	35.54	35.49
	G	39.62	37.99	33.83	38.73	39.52	41.56	41.73
	B	35.34	34.00	32.54	34.66	35.38	36.54	36.30
10	R	36.86	33.99	34.03	34.70	36.33	37.64	38.26
	G	40.86	39.17	36.15	40.00	40.23	42.19	42.64
	B	36.08	34.88	34.78	35.55	36.13	36.51	36.83
11	R	38.12	36.13	36.16	36.91	38.16	39.25	39.82
	G	40.78	39.34	37.11	40.44	40.19	41.66	42.57
	B	37.19	34.73	34.33	35.75	36.81	37.50	37.66
12	R	37.13	33.60	34.49	34.74	35.37	37.62	38.36
	G	40.17	40.09	37.66	39.59	39.70	41.45	41.49
	B	35.70	36.24	36.24	36.47	37.11	37.51	37.59
13	R	39.80	37.91	38.11	38.66	40.01	42.23	41.77
	G	43.46	42.16	39.90	42.57	43.82	45.55	44.89
	B	37.65	36.20	36.51	36.75	37.19	37.88	38.13
14	R	37.85	37.33	36.82	37.74	38.66	39.28	39.39
	G	41.37	40.65	38.79	41.13	41.93	42.62	42.84
	B	35.64	34.30	34.45	34.78	35.00	35.82	36.12
15	R	36.44	34.88	34.87	35.32	36.23	37.34	36.95
	G	41.20	40.27	38.13	40.71	40.75	42.39	42.68
	B	38.17	36.84	36.52	37.30	37.90	38.49	38.99
16	R	32.75	30.95	28.75	31.95	32.21	34.18	34.97
	G	34.09	32.36	28.60	33.22	32.99	35.00	35.59
	B	31.63	26.85	24.87	28.06	28.30	31.12	31.53
17	R	31.24	27.12	25.35	28.32	29.24	31.60	32.14
	G	35.17	32.13	26.68	33.31	33.62	37.31	37.62
	B	30.69	26.65	25.06	27.77	28.38	30.78	30.91
18	R	32.69	32.30	31.61	33.32	33.24	34.63	34.58
	G	36.20	35.69	33.84	37.02	35.91	37.30	37.27
	B	33.43	31.90	31.11	32.93	33.44	34.87	34.30
Average	R	34.71	33.05	32.68	34.06	34.71	36.10	36.23
	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
7	0.1983	0.1945	0.1459	0.1368	0.1163	0.1282	0.1397
8	0.1249	0.2336	0.1988	0.1344	0.0567	0.0510	0.0686
9	0.1387	0.2160	0.3585	0.2174	0.0915	0.0524	0.0896
10	0.1020	0.1464	0.2212	0.1526	0.0739	0.0434	0.0577
11	0.1425	0.2264	0.3280	0.2260	0.1130	0.0667	0.0801
12	0.1011	0.1835	0.2102	0.1587	0.0481	0.0243	0.0572
13	0.1249	0.2040	0.2098	0.1821	0.0577	0.0200	0.0830
14	0.1135	0.1721	0.2102	0.1649	0.0653	0.0381	0.0682
15	0.1564	0.1955	0.2975	0.2417	0.0982	0.0610	0.1120
16	0.1549	0.2150	0.3852	0.2350	0.1554	0.0920	0.1096
17	0.1788	0.2245	0.3933	0.2584	0.1468	0.1125	0.1254
18	0.1669	0.1444	0.4329	0.1759	0.0768	0.0567	0.0791
average	0.1508	0.1949	0.3107	0.2001	0.0883	0.0633	0.0891

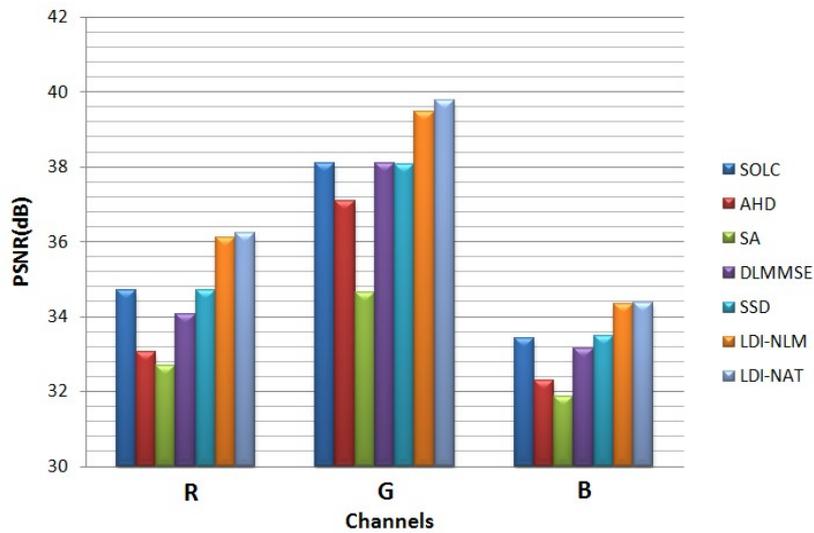


Figure 6: Graphical presentation of the average PSNR by different methods on the McMaster dataset.

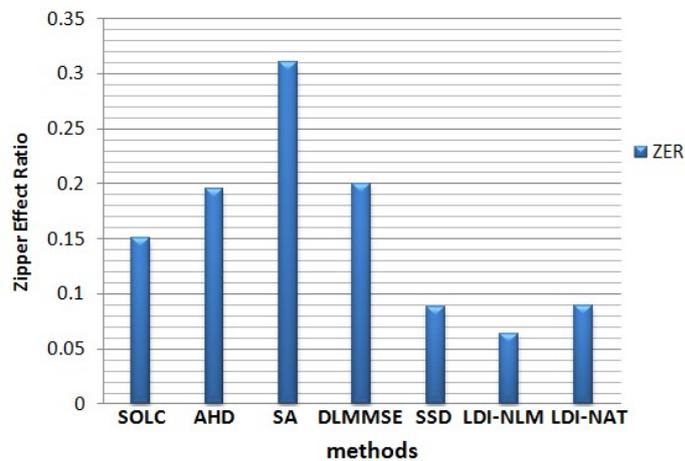


Figure 7: Graphical presentation of the average ZER by different methods on the McMaster dataset.

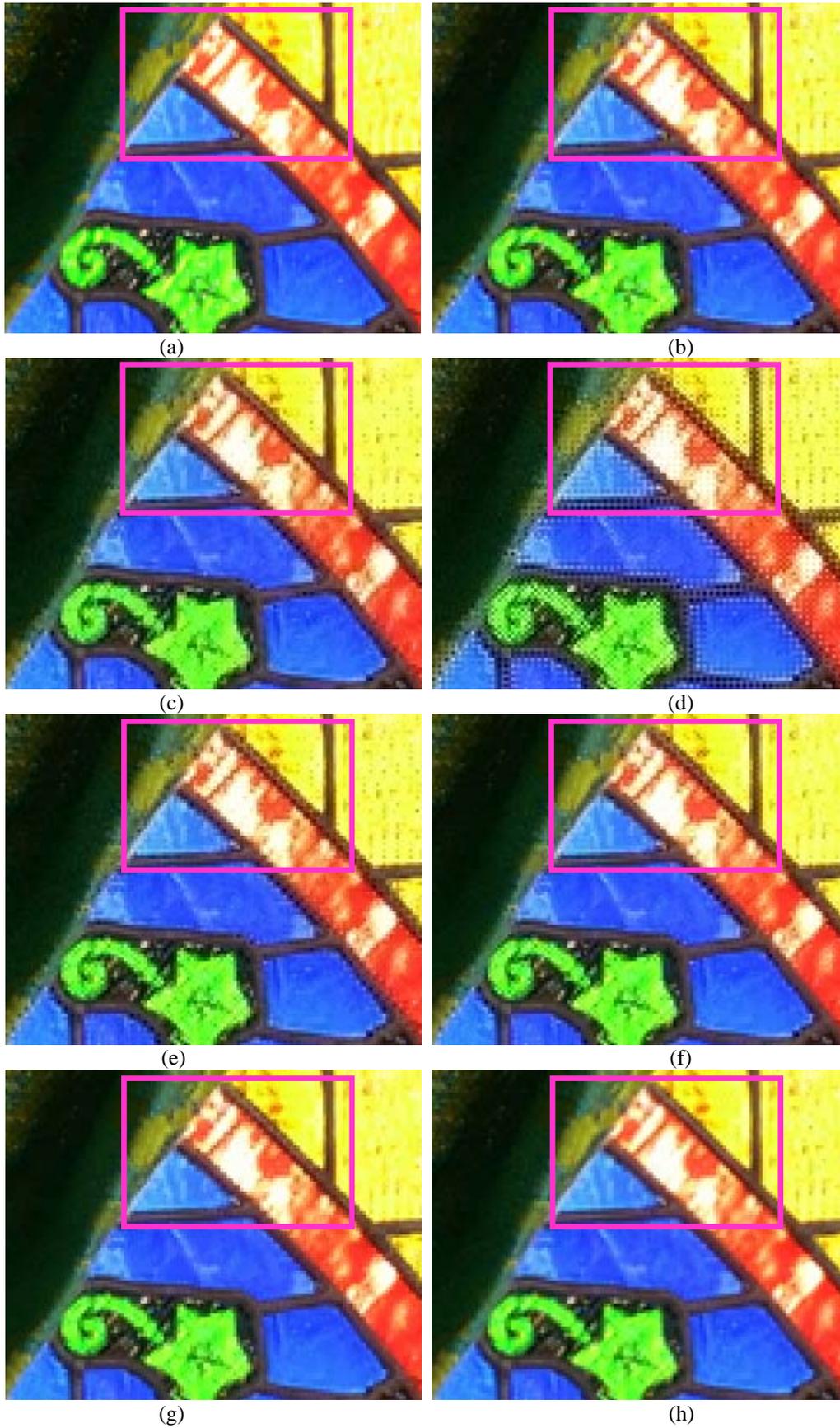


Figure 8: (a) Original image I and demosaicked images by (b) SOLC [3]; (c) AHD [10]; (d) SA [9]; (e) DLMMSE [12]; (f) SSD [15]; (g) the proposed LDI-NLM and (h) LDI-NAT.

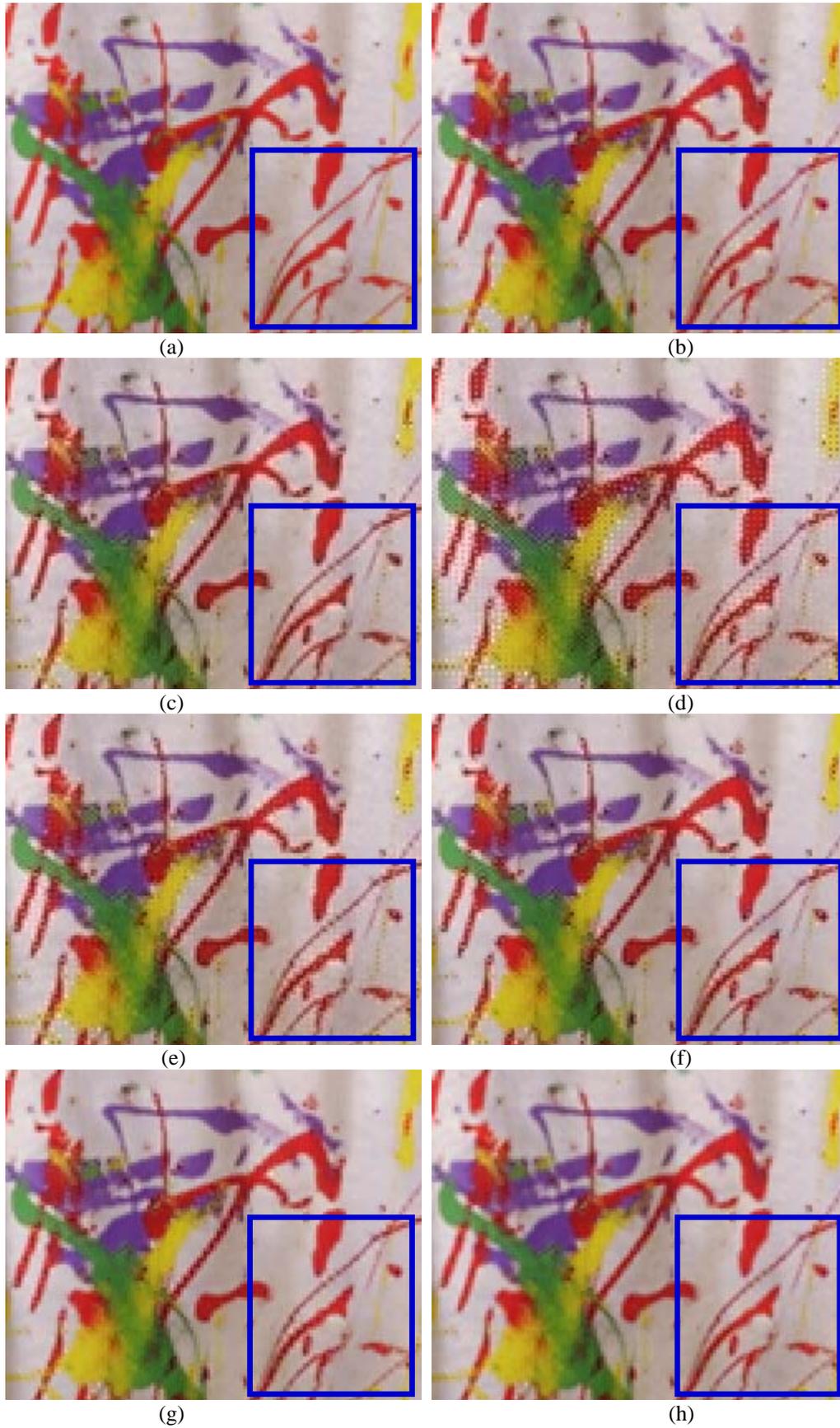


Figure 9: (a) Original image 5 and demosaicked images by (b) SOLC [3]; (c) AHD [10]; (d) SA [9]; (e) DLMMSE [12]; (f) SSD [15]; (g) the proposed LDI-NLM and (h) LDI-NAT.

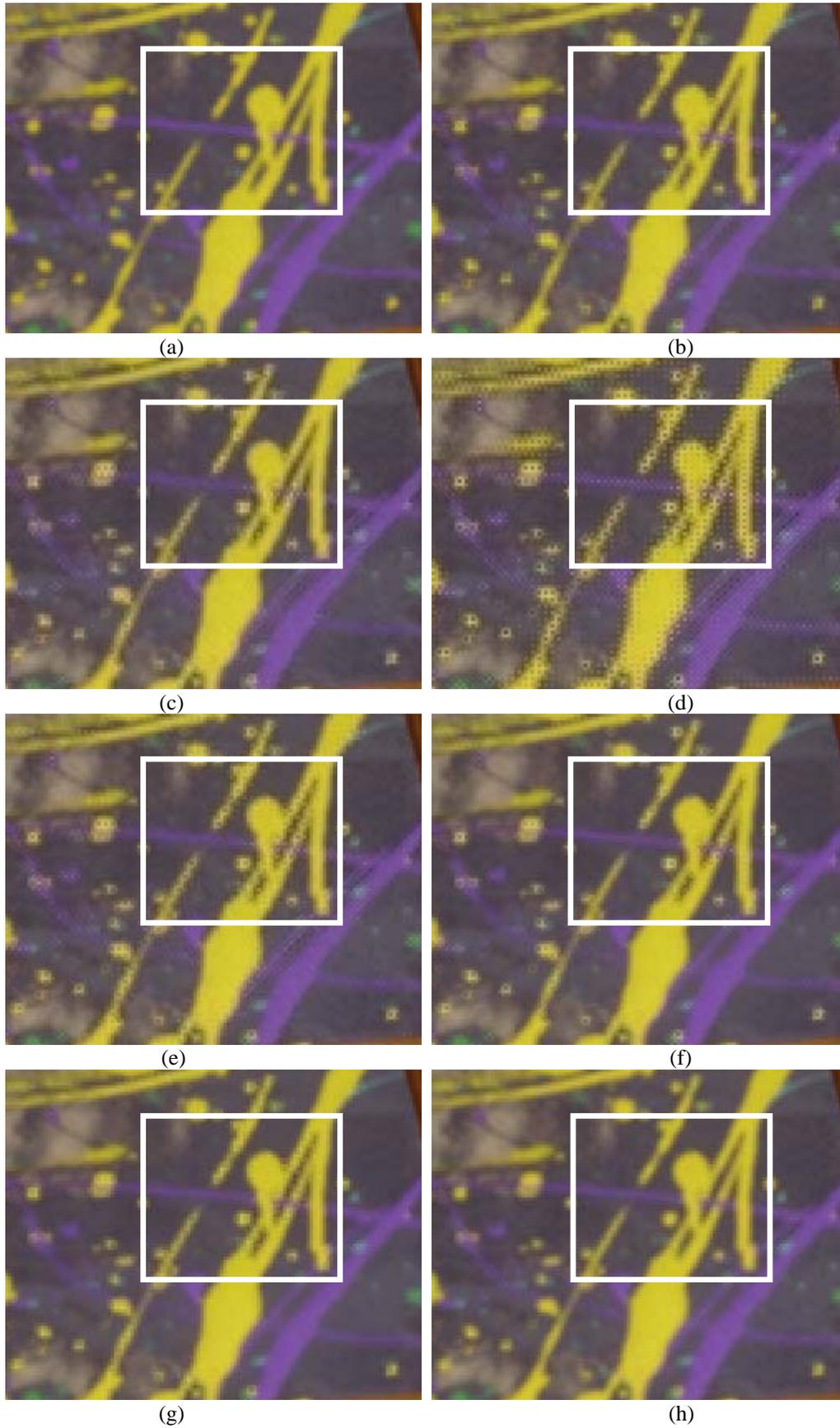


Figure 10: (a) Original image 6 and demosaicked images by (b) SOLC [3]; (c) AHD [10]; (d) SA [9]; (e) DLMMSE [12]; (f) SSD [15]; (g) the proposed LDI-NLM and (h) LDI-NAT.



Figure 11: (a) Original image *16* 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.