



Published in Image Processing On Line on 2015-12-21.
Submitted on 2015-07-07, accepted on 2015-11-17.
ISSN 2105-1232 © 2015 IPOL & the authors CC-BY-NC-SA
This article is available online with supplementary materials,
software, datasets and online demo at
<http://dx.doi.org/10.5201/ipol.2015.145>

A Demosaicking Algorithm with Adaptive Inter-Channel Correlation

Joan Duran, Antoni Buades

Universitat de les Illes Balears, Spain ([joan.duran](mailto:joan.duran@uib.es), [toni.buades](mailto:toni.buades@uib.es))

Abstract

Most common cameras use a CCD sensor device measuring a single color per pixel. Demosaicking is the interpolation process by which one can infer a full color image from such a matrix of values, thus interpolating the two missing components per pixel. Most demosaicking methods take advantage of inter-channel correlation locally selecting the best interpolation direction. The obtained results look convincing except when local geometry cannot be inferred from neighboring pixels or channel correlation is low. In these cases, these algorithms create interpolation artifacts such as zipper effect or color aliasing. This paper discusses the implementation details of the algorithm proposed in [J. Duran, A. Buades, “Self-Similarity and Spectral Correlation Adaptive Algorithm for Color Demosaicking”, IEEE Transactions on Image Processing, 23(9), pp. 4031–4040, 2014]. The proposed method involves nonlocal image self-similarity in order to reduce interpolation artifacts when local geometry is ambiguous. It further introduces a clear and intuitive manner of balancing how much channel-correlation must be taken advantage of.

Source Code

An ANSI C source code implementation of the described algorithms is accessible at the IPOL [web page of this article](#)¹, together with an on-line demo.

Keywords: image demosaicking; Bayer pattern; color filter array (CFA); nonlocal filtering; channel correlation; directional interpolation

1 Introduction

Digital color images are usually represented by three color values at each pixel: red, green, and blue. Nevertheless, most common cameras use a CCD sensor device measuring a single color per pixel. The other two components must be interpolated from neighboring pixels in the so-called demosaicking process. The selected configuration of the CCD sensor usually follows the Bayer color filter array [1], CFA. Out of a group of four pixels, two are green (in quincunx), one is red and one is blue. This configuration provides equal horizontal and vertical sampling frequency for each color. As one can see in Figure 1, the subsampling factor is 4 for red and blue channels, and 2 for the green channel.

¹<http://dx.doi.org/10.5201/ipol.2015.145>

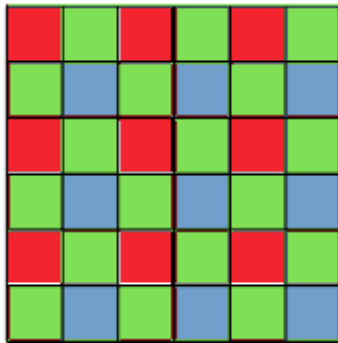


Figure 1: Bayer color filter array, CFA, which is typically used by most common cameras.

Most of state-of-the-art algorithms have been mainly tested on the Kodak database (Figure 2). Images in this base have few color saturated regions, but are challenging by their Nyquist frequency details. The large inter-channel correlation of these images explains why most methods assume that high frequencies of the three channels are highly correlated.

Recently, methods have emerged dealing with a completely different database, the IMAX collection (Figure 3), introduced by Li et al. [13]. Images in this database have many more saturated colors and edges separating colored regions than Kodak pictures. Many of these algorithms take advantage of the nonlocal self-similarity of the image [3, 16].

Most of the algorithms are designed to deal exclusively with one of the databases. That is, algorithms are conceived completely differently depending on the amount of channel-correlation in the test images being used. As a consequence, they give state-of-the-art results in one basis but usually poor results in the other one. We refer the reader to [10, 13, 14] for overviews over demosaicking methods and their relations.



Figure 2: Selected set of images of size 768×512 from Kodak dataset. Critical details are used for comparative testing in the experimental section.

We proposed in [6] a single algorithm for which inter-channel correlation is encoded in a clear and intuitive manner and automatically adapted depending on the image. The demosaicking technique reconstructs the original image by means of a two-step algorithm. First, a local directional interpolation is proposed. This local interpolation combines different directionally interpolated images depending on its chromatic smoothness. It automatically evaluates the degree of inter-channel correlation that must be taken advantage of. In a second step, a nonlocal filtering based on the NL-means Algorithm [2] is applied. This method is different from previous nonlocal demosaicking strategies since it filters the channel differences instead of the channel values themselves. This nonlocal filtering is also automatically adapted to the degree of inter-channel correlation on the initial image.



Figure 3: Selected set of images of size 400×400 from IMAX dataset. Critical details are used for comparative testing in the experimental section.

2 First Step. Local Directional Interpolation with Adaptive Inter-Channel Correlation

Because of the higher sampling rate of the green component, we first interpolate the green channel in order to better recover the main geometry and texture of the underlying true image. At each missing green position, four estimates are obtained by exploiting independently local interpolation along north, south, east and west directions. Then, for each of these directionally interpolated green components, we reconstruct red and blue values by a bilinear interpolation of the difference channels green-red and green-blue. A final decision of the most suitable approximation is made at each pixel on the basis of all reconstructed images. Importantly, all interpolation processes introduce a parameter that controls the amount of inter-channel correlation one can take advantage of according to the characteristics of the image.

In what follows, let Ω be the image domain. Furthermore, let Ω_R , Ω_G and Ω_B denote the subset of the image grid where the values of the red, green and blue channels are respectively available. We suppose that the mosaic is given by the Bayer pattern in Figure 1. We first establish how to identify the channel correlation among channels.

2.1 Identifying Inter-Channel Correlation

The strong differences between Kodak and IMAX basis can be illustrated by comparing the mean value of the chromatic gradients on several images of both collections. For this purpose, we project the RGB coordinates of each reference image into the YUV space according to the following linear combination

$$Y = \alpha_R R + \alpha_G G + \alpha_B B, \quad U = R - Y, \quad V = B - Y,$$

where $\alpha_R = 0.299$, $\alpha_G = 0.587$, and $\alpha_B = 0.114$. This color system separates the geometric information contained in Y from the chromatic information contained in U and V .

Let us define the set of pixels

$$I = \{(i, j) : |(\nabla Y)_{i,j}| > M\},$$

where $M > 0$ is a bounding parameter above which a discontinuity in the luminance gradient is considered. We then compute the L^1 -norm of the gradient of the chromatic components as

$$\begin{aligned} |\nabla U| &= \frac{1}{2|I|} \sum_{(i,j) \in I} (|U_{i+1,j} - U_{i,j}| + |U_{i,j+1} - U_{i,j}|), \\ |\nabla V| &= \frac{1}{2|I|} \sum_{(i,j) \in I} (|V_{i+1,j} - V_{i,j}| + |V_{i,j+1} - V_{i,j}|). \end{aligned} \quad (1)$$

We restrict (1) to contrasted parts of the image in order to eliminate the effect of large constant zones which are easily interpolated and carry no information about the chromatic regularity of the image. Table 1 displays the results for $M = 13$. As expected, chromatic gradients are much larger on the IMAX collection than on Kodak, which indicates that channels of Kodak images are more correlated.

Image	1	2	3	4	5	6	7	8	Avg.
Kodak	1.27	2.41	1.57	1.83	3.52	1.79	1.74	1.71	1.98
Imax	11.27	8.78	8.97	10.08	8.46	7.20	7.00	6.30	8.51

Table 1: Average of the chromatic gradients computed on the reference images in Figure 2 (Kodak database) and Figure 3 (IMAX database). The bounding parameter used in (1) is set to $M = 13$.

In order to balance the assumption on chromatic regularity for each image, we introduce a parameter β that can naturally range between zero and one, $0 < \beta \leq 1$, meaning zero that there is no correlation among channels. It is clear that even for IMAX images a correlation among channels exists. We experimentally checked that a minimal value of $\beta = 0.7$ can be used. At the end of this section, we propose a way to compute β automatically.

2.2 Interpolation of the Green Channel

Consider a CFA block as shown in Figure 1 and focus on a pixel $(i, j) \notin \Omega_G$ where the green color has to be estimated. Note that the green value is instead known on the upper, lower, right, and left pixels. These values can be used to approximate the gradient along north (n), south (s), east (e) and west (w) directions. Let us denote the intensity of the green channel by $G_{i,j}$ and of the red (blue) channel by $C_{i,j}$, with $C \in \{R, B\}$. The missing green component can be easily interpolated in each case by means of

$$\begin{aligned} \hat{G}_{i,j}^n &= G_{i,j-1} + \frac{\beta}{2} (C_{i,j} - C_{i,j-2}), \\ \hat{G}_{i,j}^s &= G_{i,j+1} + \frac{\beta}{2} (C_{i,j} - C_{i,j+2}), \\ \hat{G}_{i,j}^e &= G_{i+1,j} + \frac{\beta}{2} (C_{i,j} - C_{i+2,j}), \\ \hat{G}_{i,j}^w &= G_{i-1,j} + \frac{\beta}{2} (C_{i,j} - C_{i-2,j}). \end{aligned} \quad (2)$$

Let us somehow justify formulas in Equation (2). For the sake of simplicity, consider the interpolation process in the north direction and choose $\beta = 1$. If there is an edge along this direction, then upper neighboring pixels have similar values, that is, $\frac{1}{2} (C_{i,j} + C_{i,j-2}) \simeq C_{i,j-1}$, where $C_{i,j-1}$ stands for the real value of the red (blue) channel at $(i, j - 1)$ whenever it could be known. Consequently, the missing green value can be written as

$$\hat{G}_{i,j} = G_{i,j-1} + (C_{i,j} - C_{i,j-1}).$$

Since $C_{i,j} - C_{i,j-1}$ is a first order discrete gradient in the north direction, this procedure reconstructs the green component by imposing it to have the high frequencies of the red and blue channels, weighted by β .

2.3 Interpolation of the Red and Blue Channels

With each of the four locally interpolated green channels, we estimate the red and blue by exploiting the spectral correlation. As pointed out by Cok et al. [5], the difference images between red (blue) and green mostly contain low-frequency components and can be more accurately interpolated than the red and blue channels themselves.

Let m stand for the direction along which the green channel has been previously interpolated (north, south, east or west). Consider a pixel $(i, j) \notin \Omega_C$ where the channel $C \in \{R, B\}$ has to be estimated, and define the following vector

$$CG_{i,j}^m := C_{i,j} - \beta G_{i,j}^m.$$

Here $G_{i,j}^m$ denotes either an original CFA green sample or an interpolated one. A bilinear interpolation is then applied to estimate $CG_{i,j}^m$ according to the CFA configuration

$$\widehat{CG}_{i,j}^m = \begin{cases} \frac{CG_{i-1,j}^m + CG_{i+1,j}^m}{2} & \text{if } (i, j) \in \Omega_G \wedge (i+1, j) \in \Omega_C, \\ \frac{CG_{i,j-1}^m + CG_{i,j+1}^m}{2} & \text{if } (i, j) \in \Omega_G \wedge (i+1, j) \notin \Omega_C, \\ \frac{CG_{i-1,j-1}^m + CG_{i+1,j-1}^m + CG_{i-1,j+1}^m + CG_{i+1,j+1}^m}{4} & \text{if } (i, j) \notin \Omega_G. \end{cases} \quad (3)$$

Finally, the missing red (blue) value at each pixel is recovered as

$$\widehat{C}_{i,j}^m = \widehat{CG}_{i,j}^m + \beta G_{i,j}^m.$$

2.4 Pixel-level Fusion of Full Color Interpolated Images

The previous process provides four fully color images, each one interpolated in a different direction. In order to weight each estimate, we compute the variation of the chrominance components in the YUV space at each pixel along the four directions.

Let us denote by R^n , G^n and B^n the color components of the demosaicked image obtained from the north interpolation of the green channel and subsequent bilinear reconstruction of red and blue channels. Let U^n and V^n stand for the associated chromatic components in YUV space. Similarly, denote the vector components regarding south, east and west interpolations. For each direction, we compute the variation of the chromatic components by using a local neighborhood of L pixels in the

same direction of interpolation

$$\begin{aligned}
 \nabla_{i,j}^n &:= \frac{1}{L} \sum_{X \in \{U,V\}} \left(\sum_{l=1}^L (X_{i,j-l}^n - X_{i,j}^n)^2 \right)^{\frac{1}{2}}, \\
 \nabla_{i,j}^s &:= \frac{1}{L} \sum_{X \in \{U,V\}} \left(\sum_{l=1}^L (X_{i,j+l}^s - X_{i,j}^s)^2 \right)^{\frac{1}{2}}, \\
 \nabla_{i,j}^e &:= \frac{1}{L} \sum_{X \in \{U,V\}} \left(\sum_{l=1}^L (X_{i+l,j}^e - X_{i,j}^e)^2 \right)^{\frac{1}{2}}, \\
 \nabla_{i,j}^w &:= \frac{1}{L} \sum_{X \in \{U,V\}} \left(\sum_{l=1}^L (X_{i-l,j}^w - X_{i,j}^w)^2 \right)^{\frac{1}{2}}.
 \end{aligned} \tag{4}$$

In general terms, a large gradient along one direction means that there is an edge or image feature across it and, thus, it is suitable to avoid averaging in that direction. We simply let the weight assigned to a directional estimate to be inversely proportional to the corresponding gradient

$$\tilde{\omega}_{i,j}^m = \frac{1}{\nabla_{i,j}^m + \varepsilon}, \quad \forall m \in \{n, s, e, w\}, \tag{5}$$

where $\varepsilon > 0$ avoids dividing by too small values or even zero. The weights are normalized at each pixel as

$$\omega_{i,j}^m = \frac{\tilde{\omega}_{i,j}^m}{W_{i,j}}, \quad \forall m \in \{n, s, e, w\},$$

with $W_{i,j} = \tilde{\omega}_{i,j}^n + \tilde{\omega}_{i,j}^s + \tilde{\omega}_{i,j}^e + \tilde{\omega}_{i,j}^w$. Finally, the four directional estimates are fused in the color RGB space by means of

$$\begin{aligned}
 \hat{R}_{i,j} &= \omega_{i,j}^n R_{i,j}^n + \omega_{i,j}^s R_{i,j}^s + \omega_{i,j}^e R_{i,j}^e + \omega_{i,j}^w R_{i,j}^w, \\
 \hat{G}_{i,j} &= \omega_{i,j}^n G_{i,j}^n + \omega_{i,j}^s G_{i,j}^s + \omega_{i,j}^e G_{i,j}^e + \omega_{i,j}^w G_{i,j}^w, \\
 \hat{B}_{i,j} &= \omega_{i,j}^n B_{i,j}^n + \omega_{i,j}^s B_{i,j}^s + \omega_{i,j}^e B_{i,j}^e + \omega_{i,j}^w B_{i,j}^w.
 \end{aligned}$$

2.5 Automatic Selection of β Parameter

In order to provide an unsupervised selection of β we use a function of the chromatic gradient. As we do not dispose of the original image, the estimation will be computed using the interpolated image obtained from the previous subsection with $\beta = 1$. In view of the results displayed in Table 1 as well from some experiments performed with the proposed demosaicking method, we establish that $\beta = 0.7$ is an adapted value for images with a chromatic gradient above 3.25 while $\beta = 1.0$ is more adapted for images with a lower chromatic gradient. We used the following decreasing function of the chromatic gradient satisfying above requirements

$$\beta(t) = 1 - \frac{0.3}{1 + e^{490-150t}}, \tag{6}$$

where $t = \frac{1}{2} (|\nabla U| + |\nabla V|)$, and $|\nabla U|$ and $|\nabla V|$ are computed as in (1).

3 Second Step. Nonlocal Filtering of Channel Differences

Classical demosaicking methods, exploiting only local regularity of the image, might create color artifacts and erroneous interpolations when the interpolation direction is not accurate. Inspired by the NL-means denoising Algorithm [2], Buades et al. [3] showed that these interpolation artifacts could be eliminated by involving image self-similarity and redundancy.

Instead of directly interpolating the missing values of all three channels as in [3], we propose to first estimate the green component in terms of the difference channels $G_0 - \beta C_0$, with $C_0 \in \{R_0, B_0\}$ and $\mathbf{u}_0 = (R_0, G_0, B_0)$ being the initialization provided by the local directional algorithm introduced in Section 2. The chromatic regularity assumption of the algorithm is balanced by the same value of β . Accordingly, the averaging process for the green channel in the discrete setting reads

$$G(p) = \sum_{q \in \Omega} \omega(p, q) (G_0(q) - \beta C_0(q)) + \beta C_0(p), \quad p \notin \Omega_G, \quad (7a)$$

where $C_0 = R_0$ if $p \in \Omega_R$ and $C_0 = B_0$ if $p \in \Omega_B$. Indeed, the values of R_0 and B_0 used in the previous formula coincide with the original values since they were already available in the CFA. However, both original and interpolated green values take part in the previous average.

The interpolation of the red and blue channels take advantage of the already filled green channel. The averaging process writes

$$C(p) = \sum_{q \in \Omega} \omega(p, q) (C_0(q) - \beta G(q)) + \beta G(p), \quad p \notin \Omega_C, \quad (7b)$$

where $C = \{R, B\}$.

The weight distribution is computed on the initialization \mathbf{u}_0 and it is given in all cases by

$$\omega(p, q) = \frac{1}{\Psi(p)} \exp \left(-\frac{d(\mathbf{u}_0(p), \mathbf{u}_0(q))}{h^2} \right), \quad (7c)$$

where

$$\Psi(x) = \sum_{q \in \Omega} \exp \left(-\frac{d(\mathbf{u}_0(p), \mathbf{u}_0(q))}{h^2} \right). \quad (7d)$$

is a normalisation factor, and

$$d(\mathbf{u}_0(p), \mathbf{u}_0(q)) = \sum_{t \in B(0, \rho)} \|\mathbf{u}_0(p+t) - \mathbf{u}_0(q+t)\|^2 \quad (7e)$$

computes the distance between neighbourhoods (or patches) of size ρ around p and q . The filtering parameter h controls the decay of the exponential function and, thus, the decay of the weights as a function of the Euclidean distances.

The value of h is adapted to the chromatic gradient of the image as proposed for the β parameter. Once again, there is a clear difference between the value of h required by images with few color saturated regions ($h = 32$) and the one which better performs for images with strong edges separating saturated colors areas ($h = 1$). We use here the same bound for the chromatic gradient as for the β estimation. That is, we fix $h = 1$ for images such that $\frac{1}{2} (|\nabla U| + |\nabla V|) > 3.25$ and $h = 32$ for images with a lower chromatic gradient. We use the following continuous decreasing function satisfying the above requirements

$$h(t) = 32 - \frac{31}{1 + e^{490-150t}}, \quad (8)$$

where $t = \frac{1}{2} (|\nabla U| + |\nabla V|)$, and $|\nabla U|$ and $|\nabla V|$ are computed as in (1).

4 Algorithm Chain

The search for similarity described in (7) can be performed in the whole image. However, for computational purposes, the filtering process is restricted to pixels at a certain distance from the current pixel (*search window*). That is, the weight distribution will be zero for pixels such that $\|p - q\|_\infty > \kappa$, for a certain parameter $\kappa > 0$. In practice, the distance (7e) is measured as

$$d(\mathbf{u}_0(p), \mathbf{u}_0(q)) = \begin{cases} \sum_{t \in B(0, \rho)} \|\mathbf{u}_0(p+t) - \mathbf{u}_0(q+t)\|^2 & \text{if } \|p - q\|_\infty < \kappa, \\ \infty & \text{otherwise.} \end{cases}$$

Here $B(0, \rho)$ is a $\rho \times \rho$ discrete window centered at $(0, 0)$ (*comparison window*).

In general, the smaller the distance $d(p, q)$, the more similar the configurations of \mathbf{u}_0 around p and q . Based on this distance, only the N most similar pixels to p , q_1, \dots, q_N , take part in the average. Furthermore, the weight of the reference pixel, $\omega(p, p)$, is set to the maximum of the weights $\omega(p, \cdot)$. This setting avoids an excessive weighting of the reference point.

The values of h and β have been set experimentally and we found that they depended mainly on M , the bounding parameter for the gradient of the luminance in (1). Finally, the functions given in (6) and (8) depend only on M , while the rest of parameters such as L in (4) and ε in (5) were fixed to be in accordance with the obtained experimental results.

The final algorithm chain adapts first the values of β and h to the chromatic gradients of the image provided by the interpolation process described in Section 2. After that, the algorithm applies the proposed local directional interpolation adapted to β . The obtained demosaicked image is used as the initialization \mathbf{u}_0 for the nonlocal enhancement of the channel differences, above presented. The implementation of the proposed method is detailed in Algorithms 1-3, which have been moved to Appendix A for the sake of readability.

5 Experimental Results

We illustrate in this section the performance of the proposed method for several images in the Kodak and IMAX databases displayed in Figures 2 and 3. We compare it with the following classical and state-of-the-art demosaicking techniques: Hamilton-Adams (HA) [11], Directional Linear Minimum Mean Square-Error Image Demosaicking (DLMMSE) [15], Self-Similarity Driven Demosaicking (SSD) [3], Local Directional Interpolation with Nonlocal Adaptive Thresholding (LDNAT) [16], Image Demosaicking with Contour Stencils (CS) [7], and Minimized-Laplacian Residual Interpolation (MLRI) [12]. We implemented HA in C/C++, whereas for LDNAT and MLRI Matlab codes were downloaded from their respective webpages. Reliable software and online demos for the other approaches can be found at the IPOL [4, 8, 9]. All parameters appearing in each algorithm were selected as given in the original papers.

For the proposed method, the very same values of the parameters were fixed once and for all. The search for similar pixels is restricted to a search window of size 21×21 ($\kappa = 10$), the comparison window is a 3×3 neighborhood ($\rho = 3$), the number of similar pixels used in the nonlocal average is $N = 10$, the size of the local neighborhood used in (4) is $L = 3$, the thresholding parameter used in (5) is $\varepsilon = 10^{-8}$, and the bounding parameter for the gradient of the luminance in (1) is $M = 13$.

5.1 Numerical Quality Assessment

We evaluate the quality of the results by means of the Root Mean Squared Error (RMSE), which is a simple metric that directly takes the difference in intensity values into account. If we denote by

Kodak	HA	DLMMSE	SSD	LDNAT	CS	MLRI	Ours
1	5.14	2.63	4.17	4.36	2.36	3.66	2.30
2	2.41	1.86	2.00	2.02	1.79	1.79	1.71
3	4.50	2.33	3.56	4.03	2.51	2.83	2.30
4	4.31	2.59	3.56	4.03	2.45	3.09	2.45
5	2.08	1.77	2.04	1.99	1.84	1.65	1.85
6	2.44	1.79	2.19	2.37	1.87	1.91	1.73
7	2.96	2.00	2.62	2.80	2.15	2.24	1.99
8	3.41	2.19	2.96	3.26	2.40	2.54	2.12
Avg	3.41	2.15	2.89	3.11	2.17	2.46	2.06

(a) Kodak basis (Figure 2).

IMAX	HA	DLMMSE	SSD	LDNAT	CS	MLRI	Ours
1	9.22	10.34	9.36	7.69	7.72	7.78	7.89
2	9.19	10.52	9.61	8.02	8.52	7.90	8.01
3	5.88	6.95	5.91	4.58	4.80	4.29	4.77
4	5.38	6.88	5.31	4.37	4.29	4.59	4.42
5	5.90	7.16	5.96	4.76	5.38	4.73	5.07
6	4.37	5.22	4.32	3.62	4.01	3.66	3.86
7	5.21	5.90	5.26	4.43	5.25	4.59	4.54
8	4.77	4.91	4.65	4.26	4.66	4.13	4.55
Avg	5.55	6.43	5.60	4.64	4.96	4.63	4.79

(b) IMAX basis (Figure 3).

	HA	DLMMSE	SSD	LDNAT	CS	MLRI	Ours
Avg	4.48	4.29	4.25	3.88	3.57	3.55	3.43

(c) Total average over Kodak and IMAX basis.

Table 2: RMSE in RGB coordinates between reference images and the results provided by each demosaicking algorithm under comparison.

$\mathbf{u}^R = (u_1^R, u_2^R, u_3^R)$ the reference image and by $\mathbf{u} = (u_1, u_2, u_3)$ the demosaicked one, then

$$\text{RMSE} = \frac{1}{3} \sum_{k=1}^3 \sqrt{\frac{\sum_{p \in \Omega} (u_k(p) - u_k^R(p))^2}{|\Omega|^2}}.$$

Table 2 displays the RMSE in RGB coordinates for both Kodak and IMAX databases. In particular, Table 2a contains the results for all the pictures from Figure 2, whereas Table 2b shows the errors associated to each image from Figure 3. In general terms, all methods perform significantly better on the Kodak database than on the IMAX collection. On the one hand, DLMMSE, CS, and the proposed algorithm have a similar performance on Kodak, being the error of the first two methods larger than the one of our model in all examples. Although DLMMSE is pointed out as one of the best algorithms on Kodak, it does not work satisfactory on IMAX. On the other hand, MLRI and LDNAT give the lower RMSE for the IMAX collection and only our demosaicking technique is able to get close to them. However, LDNAT is not at all competitive for Kodak images and MLRI is far from the results obtained with the proposed method. The average RMSE over both basis in Table 2c reveals that the algorithm discussed in this paper outperforms all others.

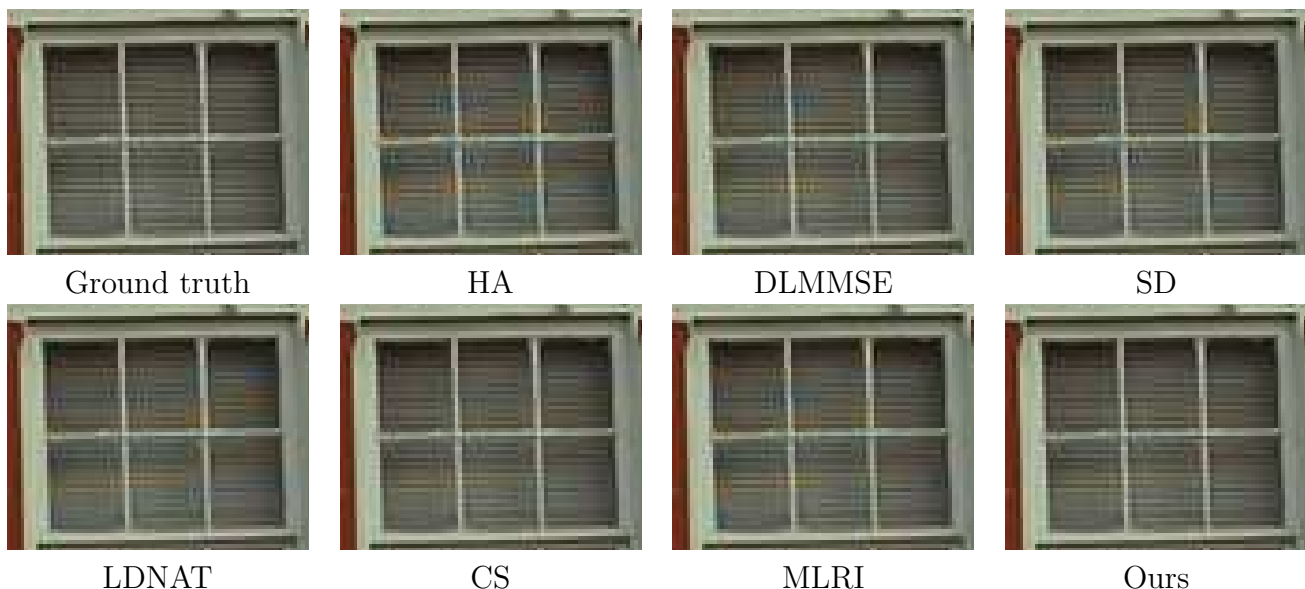


Figure 4: Test on the first Kodak image from Figure 2. This figure illustrates how the pattern of the window blinds is distorted due to wrong guesses at corners and crossing points. On the other hand, relevant color artifacts also appear in most of the demosaicked images. Only the proposed algorithm gives a fully visual acceptable solution since it preserves the original window pattern as well as it is able to reduce the color spectral distortion.

5.2 Visual Quality Assessment

A low performance in RMSE generally entails a rejection by a human visual inspection. However, any numerical criterion cannot fully replace human evaluation, which still is the most important criterion to judge the performance of demosaicking techniques. To this end, Figures 4 to 6 display the demosaicked images provided by each method on some pictures from the Kodak and IMAX collections.

Figure 4 points out how difficult it is for all demosaicking algorithms to make guesses at corners or at crossing points, even though the image is mainly gray. Note that all techniques except ours either modify the original pattern of the window blind or introduce several color artifacts. Only the proposed method gives a fully visual acceptable solution in this case.

Figure 5 reveals that most of the methods under comparison incur a high risk of creating wrong color spots in large gray regions. We observe that HA, SSD, LDNAT, CS and MLRI present color aliasing and spots since they are not able to correctly choose between horizontal and vertical interpolation. However, DLMMSE and the new-proposed algorithm reconstruct accurately the original image and the solutions illustrate significant perception gain. This means that very few color artifacts will be created by these methods in gray zones of the images. Since images in the Kodak database have very few color saturated regions, these methods have a very good performance in this basis.

Figure 6 illustrates the zipper effect on the IMAX basis. Note that DLMMSE, SSD and MLRI show significant zipper effect because of erroneous color frequency copying or because diagonal patterns are not interpolated correctly. LDNAT, CS and the proposed algorithm exhibit a higher performance in the compared visual analysis, although some zipper always remains.

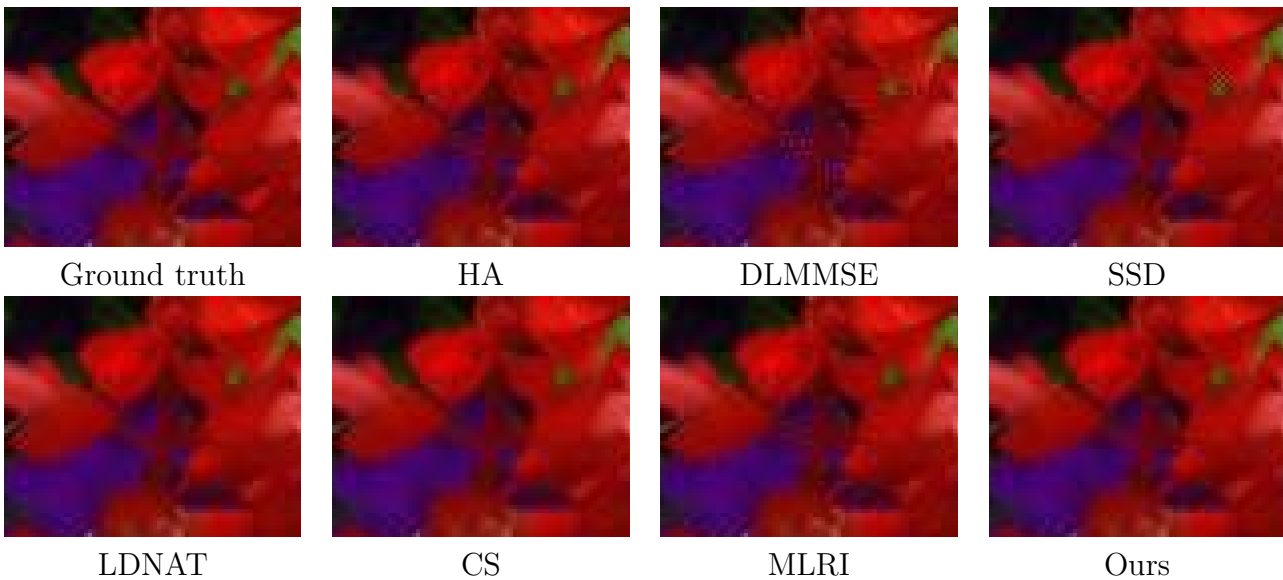
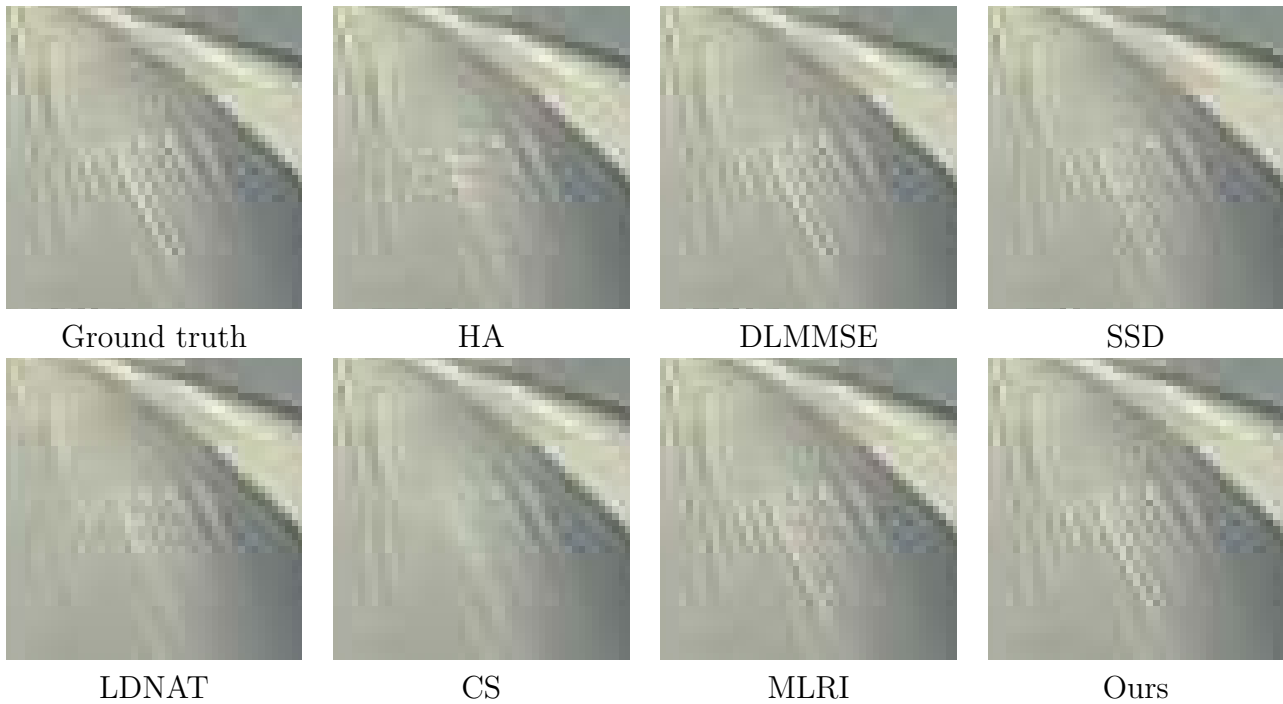


Figure 6: Test on the fourth IMAX image from Figure 3. This figure illustrates the zipper effect on color edges due to erroneous color frequency copying (this is the case of SSD) or due to erroneous interpolation of diagonal patterns (this is the case of DLMMSE and MLRI). Only LDNAT, CS and the proposed demosaicking algorithm give a reasonable solution without significant on-off image patterns.

6 Conclusions

In this paper, we analyze the implementation of the algorithm we introduced in [6] for demosaicking. The proposed method first fills the missing components by deciding *a posteriori* among four local directionally interpolated images. In a second step, a patch-based filtering is performed to reduce interpolation errors by taking advantage of image self-similarity. In both cases, the interpolation is carried out on channel differences instead of channels themselves. Interestingly, the algorithm introduces a parameter that balances how much inter-channel correlation can be used for each image.

7 On Line Demo

An online demo of the proposed algorithm is available at the IPOL [web page of this article](#)². The ANSI C source code (documented) used in this on line demo is also available.

The inputs are a color image and a value of the channel-correlation parameter $\beta \in (0, 1]$. There is also an option to estimate it automatically, as described in the paper. Users can upload their own images or use some of the images available in the demo page. If β is manually selected, then the filtering parameter is computed as $h = \frac{310\beta - 214}{3}$, according to (6) and (8).

The outputs are the uploaded data and the demosaicked image. We also include the mosaicked image, the difference image between the result and the ground truth, and the RMSE value. The difference images are computed by rescaling their range from $[-20, 20]$ to $[0, 255]$.

A Algorithmic Description of the Method

The implementation of the proposed method is detailed in Algorithms 1-3.

Acknowledgements

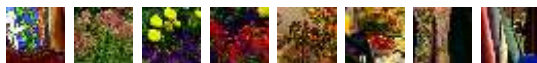
The authors would like to thank Prof. José Luis Lisani from University of Balearic Islands for the time devoted to the implementation of the on line demo.

The authors were supported by the Ministerio de Ciencia e Innovación under grant TIN2011-27539. During this work, J. Duran benefited from a fellowship of the Conselleria d'Educació, Cultura i Universitats of the Govern de les Illes Balears for the realization of his Ph.D. thesis, which was selected under an operational program co-financed by the European Social Fund.

Image Credits



by Kodak Image Suite.



by L. Zhang [13].

²<http://dx.doi.org/10.5201/ipol.2015.145>

Algorithm 1: The proposed demosaicking algorithm

input : The CFA samples (R^0, G^0, B^0) , the size of the local neighborhood $L \in \mathbb{Z}^+$, the thresholding parameter $\varepsilon > 0$, the bounding parameter for the luminance gradient $M > 0$, the half-size of the search window $\kappa \in \mathbb{Z}^+$, the size of the comparison window $\rho \in \mathbb{Z}^+$, and the number of similar pixels used in the nonlocal average $N \in \mathbb{Z}^+$.

output: The final demosaicked image (R, G, B) .

begin

$(R^1, G^1, B^1) \leftarrow \text{local_int}(R^0, G^0, B^0, 1, L, \varepsilon)$ *Estimation of β and h if not provided*

for $(i, j) \in \Omega$ **do**

$Y_{i,j} \leftarrow 0.299R_{i,j}^1 + 0.587G_{i,j}^1 + 0.114B_{i,j}^1$

$U_{i,j} \leftarrow R_{i,j}^1 - Y_{i,j}$

$V_{i,j} \leftarrow B_{i,j}^1 - Y_{i,j}$

$t \leftarrow 0$

$k \leftarrow 0$

for $(i, j) \in \Omega$ **do**

if $|Y_{i+1,j} - Y_{i,j}| + |Y_{i,j+1} - Y_{i,j}| > M$ **then**

$t \leftarrow t + \frac{1}{4}(|U_{i+1,j} - U_{i,j}| + |U_{i,j+1} - U_{i,j}| + |V_{i+1,j} - V_{i,j}| + |V_{i,j+1} - V_{i,j}|)$

$k \leftarrow k + 1$

$t \leftarrow \frac{t}{k}$

$\beta \leftarrow 1 - \frac{0.3}{1 + e^{490-150t}}$

$h \leftarrow 32 - \frac{31}{1 + e^{490-150t}}$

Initialization by local directional interpolation

$(R^2, G^2, B^2) \leftarrow \text{local_int}(R^0, G^0, B^0, \beta, L, \varepsilon)$

Final demosaicked image by nonlocal interpolation

$(R, G, B) \leftarrow \text{nonlocal_int}(R^2, G^2, B^2, \beta, h, \kappa, \rho, N)$

return (R, G, B)

Algorithm 2: `local_int` ($R^0, G^0, B^0, \beta, L, \varepsilon$) – Local directional interpolation algorithm

input : The CFA samples (R^0, G^0, B^0) , the channel-correlation parameter $\beta \in (0, 1]$, the size of the local neighborhood $L \in \mathbb{Z}^+$, and the thresholding parameter $\varepsilon > 0$.

output: The interpolated image (R, G, B) .

begin

for $m \in \{n, s, e, w\}$ **do**

Directional interpolation of green channel

for $(i, j) \in \Omega$ **do**

if $(i, j) \in \Omega_G$ **then**

$G_{i,j}^m \leftarrow G_{i,j}^0$

else

$$G_{i,j}^m \leftarrow \begin{cases} G_{i,j-1} + \frac{\beta}{2} (C_{i,j} - C_{i,j-2}) & \text{if } m = n, \\ G_{i,j+1} + \frac{\beta}{2} (C_{i,j} - C_{i,j+2}) & \text{if } m = s, \\ G_{i+1,j} + \frac{\beta}{2} (C_{i,j} - C_{i+2,j}) & \text{if } m = e, \\ G_{i-1,j} + \frac{\beta}{2} (C_{i,j} - C_{i-2,j}) & \text{if } m = w. \end{cases}$$

Bilinear interpolation of red and blue channels

for $(i, j) \in \Omega_C$ **do**

$C_{i,j}^m \leftarrow C_{i,j}^0$

$CG_{i,j}^m \leftarrow C_{i,j} - \beta G_{i,j}^m$

for $(i, j) \notin \Omega_C$ **do**

$$C_{i,j}^m \leftarrow \left\{ \begin{array}{ll} \frac{CG_{i-1,j}^m + CG_{i+1,j}^m}{2} & \text{if } (i, j) \in \Omega_G \text{ and } (i+1, j) \in \Omega_C \\ \frac{CG_{i,j-1}^m + CG_{i,j+1}^m}{2} & \text{if } (i, j) \in \Omega_G \text{ and } (i+1, j) \notin \Omega_C \\ \frac{CG_{i-1,j-1}^m + CG_{i+1,j-1}^m + CG_{i-1,j+1}^m + CG_{i+1,j+1}^m}{4} & \text{if } (i, j) \notin \Omega_G \end{array} \right\} + \beta G_{i,j}^m$$

Pixel-level fusion of full color interpolated images

for $(i, j) \in \Omega$ **do**

$\nabla_{i,j}^n, \nabla_{i,j}^s, \nabla_{i,j}^e, \nabla_{i,j}^w \leftarrow (4)$

for $m \in \{n, s, e, w\}$ **do**

$\omega_{i,j}^m \leftarrow \frac{1}{\nabla_{i,j}^m + \varepsilon}$

$W_{i,j} \leftarrow \omega_{i,j}^n + \omega_{i,j}^s + \omega_{i,j}^e + \omega_{i,j}^w$

for $m \in \{n, s, e, w\}$ **do**

$\omega_{i,j}^m \leftarrow \frac{\omega_{i,j}^m}{W_{i,j}}$

$R_{i,j} \leftarrow \omega_{i,j}^n R_{i,j}^n + \omega_{i,j}^s R_{i,j}^s + \omega_{i,j}^e R_{i,j}^e + \omega_{i,j}^w R_{i,j}^w$

$G_{i,j} \leftarrow \omega_{i,j}^n G_{i,j}^n + \omega_{i,j}^s G_{i,j}^s + \omega_{i,j}^e G_{i,j}^e + \omega_{i,j}^w G_{i,j}^w$

$B_{i,j} \leftarrow \omega_{i,j}^n B_{i,j}^n + \omega_{i,j}^s B_{i,j}^s + \omega_{i,j}^e B_{i,j}^e + \omega_{i,j}^w B_{i,j}^w$

return (R, G, B)

Algorithm 3: `nonlocal_int` ($R^0, G^0, B^0, \beta, h, \kappa, \rho, N$) – Nonlocal interpolation algorithm

input : An initialization $\mathbf{u}^0 = (R^0, G^0, B^0)$, the channel-correlation parameter $\beta \in (0, 1]$, the filtering parameter $h > 0$, the half-size of the search window $\kappa \in \mathbb{Z}^+$, the size of the comparison window $\rho \in \mathbb{Z}^+$, and the number of similar pixels used in the nonlocal average $N \in \mathbb{Z}^+$.

output: The final demosaicked image (R, G, B) .

begin

Compute weight distribution on initialization \mathbf{u}_0

for $p \in \Omega$ **do**

for $q \in \Omega, q \neq p$ **do**

$d(p, q) \leftarrow \infty$

$\omega(p, q) \leftarrow 0$

if $\|p - q\|_\infty \leq \kappa$ **then**

$d(p, q) \leftarrow \sum_{t \in B(0, \rho)} \|\mathbf{u}^0(p + t) - \mathbf{u}^0(q + t)\|^2$

$d(p, p) \leftarrow \min\{d(p, q) : q \in \Omega, q \neq p\}$

 Sort $d(p, \cdot)$ by increasing order: $d(p, q_1) \leq d(p, q_2) \leq \dots \leq d(p, q_N) \leq \dots$

for $n \in \{1, \dots, N\}$ **do**

$\omega(p, q_n) \leftarrow \exp\left(\frac{-d(p, q_n)}{h^2}\right)$

$\Psi(p) \leftarrow \sum_{n=1}^N \omega(p, q_n)$

for $n \in \{1, \dots, N\}$ **do**

$\omega(p, q_n) \leftarrow \frac{\omega(p, q_n)}{\Psi(p)}$

Enhancement of green channel

for $p \in \Omega$ **do**

if $p \in \Omega_G$ **then**

$G(p) = G^0(p)$

else

$G(p) \leftarrow \sum_{n=1}^N \omega(p, q_n) (G^0(q_n) - \beta C^0(q_n)) + \beta C^0(p)$

Enhancement of red and blue channels

for $C \in \{R, B\}$ **do**

for $p \in \Omega$ **do**

if $p \in \Omega_C$ **then**

$C(p) = C^0(p)$

else

$C(p) \leftarrow \sum_{n=1}^N \omega(p, q_n) (C^0(q_n) - \beta G(q_n)) + \beta G(p)$

return (R, G, B)

References

- [1] B.E. BAYER, *Color imaging array*, 1976. US Patent 3 971 065.
- [2] A. BUADES, B. COLL, AND J.-M. MOREL, *A review of image denoising algorithms, with a new one*, SIAM Multiscale Modeling and Simulation, 4 (2005), pp. 490–530. <http://dx.doi.org/10.1137/040616024>.
- [3] A. BUADES, B. COLL, J.-M. MOREL, AND C. SBERT, *Self-similarity driven color demosaicking*, IEEE Transactions on Image Processing, 18 (2009), pp. 1192–1202. <http://dx.doi.org/10.1109/TIP.2009.2017171>.
- [4] —, *Self-similarity driven demosaicking*, Image Processing On Line, (2011). <http://dx.doi.org/10.5201/ipol.2011.bcms-ssdd>.
- [5] D.R. COK, *Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal*, 1987. US Patent 4 642 678.
- [6] J. DURAN AND A. BUADES, *Self-similarity and spectral correlation adaptive algorithm for color demosaicking*, IEEE Transactions on Image Processing, 23 (2014), pp. 4031–4040. <http://dx.doi.org/10.1109/TIP.2014.2341928>.
- [7] P. GETREUER, *Color demosaicing with contour stencils*, in Proceedings of the 17th IEEE International Conference on Digital Signal Processing (DSP), 2011, pp. 1–6. <http://dx.doi.org/10.1109/ICDSP.2011.6004885>.
- [8] —, *Zhang-Wu directional LMMSE image demosaicking*, Image Processing On Line, (2011). http://dx.doi.org/10.5201/ipol.2011.g_zwld.
- [9] —, *Image demosaicking with contour stencils*, Image Processing On Line, (2012). <http://dx.doi.org/10.5201/ipol.2012.g-dwcs>.
- [10] B. GUNTURK, J. GLOTZBACH, Y. ALTUNBASAK, R. SCHAFER, AND R. MERSEREAU, *Demosaicking: Color filter array interpolation*, IEEE Signal Processing Magazine, 22 (2005), pp. 44–54. <http://dx.doi.org/10.1109/MSP.2005.1407714>.
- [11] J.F. HAMILTON AND J.E. ADAMS, *Adaptive color plan interpolation in single sensor color electronic camera*, 1997. US Patent 5 629 734.
- [12] D. KIKU, Y. MONNO, M. TANAKA, AND M. OKUTOMI, *Minimized-Laplacian residual interpolation for color image demosaicking*, in Proceedings of SPIE, Digital Photography, San Francisco, CA, USA, 2014, International Society for Optics and Photonics, pp. 90230L–90230L. <http://dx.doi.org/10.1117/12.2038425>.
- [13] X. LI, B. GUNTURK, AND L. ZHANG, *Image demosaicing: A systematic survey*, in Proceedings of SPIE, Visual Communications and Image Processing, vol. 6822, San Jose, CA, USA, 2008, pp. 68221J–68221J–15. <http://dx.doi.org/10.1117/12.766768>.
- [14] D. MENON AND G. CALVAGNO, *Color image demosaicking: An overview*, Signal Processing: Image Communication, 26 (2011), pp. 518–533. <http://dx.doi.org/10.1016/j.image.2011.04.003>.

- [15] L. ZHANG AND X. WU, *Color demosaicking via directional linear minimum mean square-error estimation*, IEEE Transactions on Image Processing, 14 (2005), pp. 2167–2178. <http://dx.doi.org/10.1109/TIP.2005.857260>.
- [16] L. ZHANG, X. WU, A. BUADES, AND X. LI, *Color demosaicking by local directional interpolation and nonlocal adaptive thresholding*, Journal of Electronic Imaging, 20 (2011), p. 023016. <http://dx.doi.org/10.1117/1.3600632>.