

A Fast Algorithm for the Demosaicing Problem Concerning the Bayer Pattern

Antonio Boccuto, Ivan Gerace, Valentina Giorgetti and Matteo Rinaldi

Laboratorio di Matematica Computazionale “Sauro Tulipani”
Dipartimento di Matematica e Informatica
Università degli Studi di Perugia
Via Vanvitelli, 1
I-06123 Perugia (Italy)



July 11, 2018

Abstract

In this paper we deal with the demosaicing problem when the Bayer pattern is used. We propose a fast heuristic algorithm, consisting of three parts. In the first one, we initialize the green channel by means of an edge-directed and weighted average technique. In the second part, the red and blue channels are updated, thanks to an equality constraint on the second derivatives. The third part consists of a constant-hue-based interpolation. We show experimentally how the proposed algorithm gives in mean better reconstructions than more computationally expensive algorithms.

Key words: Demosaicing, sparse data problem, inverse problem, edge-preserving image reconstruction, local filtering, Bayer Pattern.

1 Introduction

The demosaicing problem is related to the acquisition of RGB color images by means of CCD digital cameras. In the RGB model, each pixel of a digital color image is associated to a triple of numbers, which indicate the light intensity of the red, green and blue channel, respectively. However, most cameras use a single sensor, associated with a color filter that allows only the measurement at each pixel of the reflectance of the scene at one of

the three colors, according to a given scheme or pattern, called *Color Filter Array* (CFA). For this reason, at each pixel, the other two missing colors should be estimated. Different CFA's are proposed for the acquisition (see also [3, 14, 18]). The most common is the Bayer pattern (see also [4]). In this scheme, the numbers of pixels where the green color is sampled is twice what those associated with the red and blue channels are, because of the higher sensibility of the human eye to the green wavelengths. If we decompose the acquired image in three channels, we obtain three downsampled grayscale images, so that demosaicing could be interpreted as the problem of interpolating grayscale images from sparse data. In most cameras, demosaicing is a part of the processing required to obtain a visible images. The camera's built-in-firmware is substantially based on fast local interpolation algorithms.

The heuristic approaches, which do not try to solve an optimization problem defined in mathematical terms, are widely used in the literature. These methods have the advantage of being very fast. Our proposed technique is one of heuristic kind. In general, the heuristic techniques consist of filtering operations, which are formulated by means of suitable observations on color images. The nonadaptive algorithms, among which bilinear and bicubic interpolation, yield satisfactory results in smooth regions of an image, but they can fail in textured or edge areas. Edge-directed interpolation is an adaptive approach, where, by analyzing the area around each pixel, we choose the possible interpolation direction. In practice, the interpolation direction is chosen to avoid interpolating across the edges. In [15], for each pixel the horizontal and vertical gradients are compared with a constant threshold. If the gradient in one direction is greater than the threshold, then interpolation is not performed along this direction. Some other direct interpolation methods use larger neighborhoods by examining different color channels. In [23], to determine the edges of the green channels, the red and blue channels are employed. On the other hand, to determine the edges of the red and blue channels, some discrete derivation operators of the second order are used, while in [19], to determine the edges in the various channels, a suitable Jacobian operator is applied. In [16], local homogeneity is used as an indicator to choose horizontally or vertically interpolated intensities. Thanks to homogeneity-directed interpolation, the luminance and chrominance values have to be similar in a suitable neighborhood. In demosaicing it is often assumed that the differences or the ratios of the intensity values in different channels are locally constant (see also [1, 10, 15, 21, 23, 30, 32, 34]). In [21] the probability of having an edge in a certain direction is determined and used to find the weights relative to the weighted average employed as an interpolation operator. In this algorithm, the color channels are updated iteratively according to the constant color ratio condition. In [24] a similar algorithm is proposed, where 7-size neighborhoods are employed to find the edges of the green channel, and 5×5 -size neighborhoods are used to determine the edges of the red and blue channels. An analogous algorithm is defined in [35], where the interpolation can be done also in the diagonal direction, while in [33] the weighted directional interpolation is used by means of a fuzzy membership assignment. In [2] a second order operator is employed as a correction term.

To have more accurate results, several techniques, which use iterative methods, are proposed. However, they have a higher computational cost with respect to the heuristic techniques. One of well-known techniques is the algorithm of *Alternate Projections* (AP) (see [12]), which uses the strong correlation between the high frequencies of the three

colored components, by projecting alternatively the estimated image in a constraint of observation and in a constraint which imposes similarity between the red and green edges and between the blue and green edges, until a fixed point is found. Another widely used technique is regularization (see also [11, 27]). The algorithm in [20] is based on interpolation in a residual domain. The residuals are the differences between the observed and estimated pixel values which minimize a Laplacian energy.

The algorithm here presented consists of three steps. The first two ones are initialization steps, while the third one is an iterative steps. In the first one, the missing values in the green component are determined, in particular a weighted average-type technique is used. The weights are determined in an edge-directed approach, in which we consider also the possible edges in the red and blue components. In the second step, we determine the missing values in the red and blue components. In this case we use two alternative techniques according to the position in the Bayer pattern of the involved pixel. In the first technique, the missing value is determined by imposing that the second derivative of the intensity value of the red/blue channel is equal to the second derivative of the intensity values of the green channel. This is done according to the proposed approaches in the AP algorithm and the regularization algorithm given in [11]. In particular, in [11] a constraint is imposed, to get the derivatives of all channels similar as soon as possible. At the third step, all values of the three channels are recursively updated, by means of a constant-hue-based technique. In particular, we assume the constant color difference. The technique we propose at this step is similar to that used by W. T. Freeman in [10]. Indeed, even here a median filter is employed, in order to correct small spurious imperfections. We repeat iteratively the third step. However, to avoid increasing excessively the computational cost, we experimentally estimate that only four iterations are necessary to obtain an accurate demosaicing. We call our technique as *Local Edge Preserving* (LEP) algorithm.

The paper is structured as follows. In Section 2 we give a mathematical formulation of the demosaicing problem. In Section 3 we describe the initialization of the proposed algorithm, which consists of the two first steps aforementioned. In Section 4 we give the third iterative step of our algorithm, highlighting the differences with the Freeman filter. In Section 5 our experimental results are presented. This section consists of two parts. In the first one, we determine the best detection function which can be used in order to evaluate the edges. In the second one, we compare our algorithm with other techniques recently proposed in the literature and we show how the LEP method gives in mean more accurate reconstructions than the other considered algorithms.

2 The demosaicing problem

An RGB (red-green-blue) *color image* is a vector of the type

$$\mathbf{x} = \begin{pmatrix} \mathbf{r} \\ \mathbf{g} \\ \mathbf{b} \end{pmatrix} \in \mathbb{R}^{3n \cdot m},$$

where \mathbf{r} , \mathbf{g} , $\mathbf{b} \in \mathbb{R}^{n \cdot m}$ are the red, green and blue channels according to the lexicographic order, respectively. We consider the problem of acquisition of data from a digital camera, and call it *mosaicing problem*. Given an ideal image $\mathbf{x} \in \mathbb{R}^{3n \cdot m}$, the acquired or *mosaiced*

image is defined by

$$\bar{\mathbf{x}} = \begin{pmatrix} \bar{\mathbf{r}} \\ \bar{\mathbf{g}} \\ \bar{\mathbf{b}} \end{pmatrix} = M\mathbf{x},$$

where $\bar{\mathbf{x}} \in \mathbb{R}^{3n \cdot m}$ and $M \in \mathbb{R}^{(3n \cdot m) \times (3n \cdot m)}$ is a linear operator defined by setting

$$M = \begin{pmatrix} M^{(r)} & O & O \\ O & M^{(g)} & O \\ O & O & M^{(b)} \end{pmatrix},$$

where $O \in \mathbb{R}^{(n \cdot m) \times (n \cdot m)}$ is the null matrix, and $M^{(r)}, M^{(g)}, M^{(b)} \in \mathbb{R}^{(n \cdot m) \times (n \cdot m)}$ are diagonal matrices whose principal entries, if we use the Bayer pattern (see Figure ??), are given by

$$\begin{aligned} m_{(i,j),(i,j)}^{(r)} &= \begin{cases} 1, & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 0, \\ 0, & \text{otherwise;} \end{cases} \\ m_{(i,j),(i,j)}^{(g)} &= \begin{cases} 1, & \text{if } i \not\equiv_2 j, \\ 0, & \text{otherwise;} \end{cases} \\ m_{(i,j),(i,j)}^{(b)} &= \begin{cases} 1, & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 1, \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

R ₀₀	G ₀₁	R ₀₂	G ₀₃	R ₀₄	G ₀₅	R ₀₆
G ₁₀	B ₁₁	G ₁₂	B ₁₃	G ₁₄	B ₁₅	G ₁₆
R ₂₀	G ₂₁	R ₂₂	G ₂₃	R ₂₄	G ₂₅	R ₂₆
G ₃₀	B ₃₁	G ₃₂	B ₃₃	G ₃₄	B ₃₅	G ₃₆
R ₄₀	G ₄₁	R ₄₂	G ₄₃	R ₄₄	G ₄₅	R ₄₆
G ₅₀	B ₅₁	G ₅₂	B ₅₃	G ₅₄	B ₅₅	G ₅₆
R ₆₀	G ₆₁	R ₆₂	G ₆₃	R ₆₄	G ₆₅	R ₆₆

Figure 1: Bayer Pattern.

The corresponding demosaicing problem is the associated inverse problem, that is to determine the ideal color image \mathbf{x} , knowing the mosaiced image $\bar{\mathbf{x}}$ and the linear operator M . An inverse problem is said to be *well-posed* (in the sense of Hadamard) if and only if the solution exists, is unique and stable with respect to data variation. A not well-posed problem is said to be *ill-posed* (see also [13]). Note that the demosaicing problem is ill-posed, since the matrix M in (1) is singular, as it is readily seen, and so there are infinitely many solutions.

3 The initialization of the proposed algorithm

In the initialization phase we proceed as follows: first we initialize the green channel, since in the green channel we have more data than in the other ones, and successively, thanks to the initialization of the green channel, we update the other two.

3.1 The initialization of the green channel

We refer to a *clique* as a pair of adjacent pixels. Every missing value of the green channel is initialized by a weighted mean of the known green values in its neighborhood. The weights of the considered mean take into account possible discontinuities in a set of adjacent cliques. We consider cliques both in the blue and in the red channel, since it is well-known that there is a correlation between the discontinuities in the various channels related to edges, such as object borders and textures (see e.g. [11]).

Here we distinguish three cases: the first one is when we have the value of the green light intensity on a pixel; the second one is when we the blue value of the involved pixel is known, that is when i and j are both odd; the third one is when the red value on the considered pixel is known, namely when i and j are both even.

The first approximation $g^{(0)}$ of the green ideal image g is given by

$$g^{(0)}_{(i,j)} = \begin{cases} \bar{g}_{(i,j)} & \text{if } i \not\equiv_2 j, \\ \frac{t_1 \bar{g}_{(i-1,j)} + t_2 \bar{g}_{(i+1,j)} + t_3 \bar{g}_{(i,j-1)} + t_4 \bar{g}_{(i,j+1)}}{t_1 + t_2 + t_3 + t_4} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 1, \\ \frac{t_5 \bar{g}_{(i-1,j)} + t_6 \bar{g}_{(i+1,j)} + t_7 \bar{g}_{(i,j-1)} + t_8 \bar{g}_{(i,j+1)}}{t_5 + t_6 + t_7 + t_8} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 0. \end{cases}$$

Note that, in the first case, we keep the value we already have. In the second case, we do a weighted mean of the intensity values taken on the adjacent pixels where the green value is known. The weights t_1, t_2, t_3, t_4 of the mean are computed by using the green and the blue channels. In particular,

$$t_1 = \phi(\tau_1), \quad (1)$$

where ϕ is a suitable positive decreasing *detection function* and τ_1 is defined by

$$\tau_1 = |\bar{g}_{(i-1,j)} - \bar{g}_{(i,j-1)}| + |\bar{g}_{(i-1,j)} - \bar{g}_{(i,j+1)}| + |\bar{b}_{(i,j)} - \bar{b}_{(i-2,j)}|.$$

When the differences between the green values on the pixels $(i-1, j)$ and $(i, j-1)$, $(i-1, j)$ and $(i, j+1)$, and between the blue values on the pixels (i, j) and $(i-2, j)$, are small enough, then we can assume that there are no discontinuities between the pixels $(i-1, j)$ and (i, j) . So, in the calculus of the green value on the pixel (i, j) , we give a large weight t_1 to the green value in the position $(i-1, j)$. Thus, when the value τ_1 is small, the probability of having a discontinuity between the pixels (i, j) and $(i-1, j)$ in the green channel is large, and vice versa.

Moreover, $t_2 = \phi(\tau_2)$, where τ_2 is given by

$$\tau_2 = |\bar{g}_{(i+1,j)} - \bar{g}_{(i,j-1)}| + |\bar{g}_{(i+1,j)} - \bar{g}_{(i,j+1)}| + |\bar{b}_{(i,j)} - \bar{b}_{(i+2,j)}|.$$

When the differences between the green values on the pixels $(i+1, j)$ and $(i, j-1)$, $(i+1, j)$ and $(i, j+1)$, and between the blue values on the pixels (i, j) and $(i+2, j)$, are sufficiently small, then we can suppose that there exist no edges between the pixels $(i+1, j)$ and (i, j) . Thus, in the calculus of the green value on the pixel (i, j) , we give a large weight t_2 to the green value in the position $(i+1, j)$.

Furthermore, $t_3 = \phi(\tau_3)$, where

$$\tau_3 = |\bar{g}_{(i-1,j)} - \bar{g}_{(i,j-1)}| + |\bar{g}_{(i,j-1)} - \bar{g}_{(i+1,j)}| + |\bar{b}_{(i,j)} - \bar{b}_{(i,j-2)}|.$$

When the differences between the green values on the pixels $(i-1, j)$ and $(i, j-1)$, $(i, j-1)$ and $(i+1, j)$, and between the blue values on the pixels (i, j) and $(i, j-2)$, are small, then we assume that there are no discontinuities between the pixels (i, j) and $(i, j-1)$. So, in the calculus of the green value on the pixel (i, j) , we have a large weight t_3 for the green value in the position $(i, j-1)$.

Finally, $t_4 = \phi(\tau_4)$, where

$$\tau_4 = |\bar{g}_{(i,j+1)} - \bar{g}_{(i-1,j)}| + |\bar{g}_{(i,j+1)} - \bar{g}_{(i+1,j)}| + |\bar{b}_{(i,j)} - \bar{b}_{(i,j+2)}|.$$

When the differences between the green values on the pixels $(i, j+1)$ and $(i-1, j)$, $(i, j+1)$ and $(i+1, j)$, and between the blue values on the pixels (i, j) and $(i, j+2)$, are small enough, then we can assume that there are no edges between the pixels $(i, j+1)$ and (i, j) . So, in the calculus of the green value on the pixel (i, j) , we give a large weight t_4 to the green value in the position $(i, j+1)$.

Even in the third case, we compute the weighted mean of the intensity values taken on the adjacent pixels where the green value is known. The weights t_5, t_6, t_7, t_8 of the mean are computed by using the green and the red channels. In particular, $t_5 = \phi(\tau_5)$, where

$$\tau_5 = |\bar{g}_{(i-1,j)} - \bar{g}_{(i,j-1)}| + |\bar{g}_{(i-1,j)} - \bar{g}_{(i,j+1)}| + |\bar{r}_{(i,j)} - \bar{r}_{(i-2,j)}|.$$

We argue analogously as in the computation of the weight t_1 , where the role of the blue channel is played by the red component.

Moreover, proceeding analogously as in the calculation of t_2 , we get $t_6 = \phi(\tau_6)$, where

$$\tau_6 = |\bar{g}_{(i+1,j)} - \bar{g}_{(i,j-1)}| + |\bar{g}_{(i+1,j)} - \bar{g}_{(i,j+1)}| + |\bar{r}_{(i,j)} - \bar{r}_{(i+2,j)}|.$$

By arguing analogously as in the computation of t_3 , we obtain $t_7 = \phi(\tau_7)$, where

$$\tau_7 = |\bar{g}_{(i-1,j)} - \bar{g}_{(i,j-1)}| + |\bar{g}_{(i,j-1)} - \bar{g}_{(i+1,j)}| + |\bar{r}_{(i,j)} - \bar{r}_{(i,j-2)}|.$$

Finally, by proceeding as in the computation of t_4 , one has $t_8 = \phi(\tau_8)$, where

$$\tau_8 = |\bar{g}_{(i,j+1)} - \bar{g}_{(i-1,j)}| + |\bar{g}_{(i,j+1)} - \bar{g}_{(i+1,j)}| + |\bar{r}_{(i,j)} - \bar{r}_{(i,j+2)}|.$$

3.2 The initialization of the red values

Here we distinguish four cases: the first one is when we already know the red value of a pixel; the second one is when we know the red values in the two adjacent pixels in the same column, that is i is odd and j is even (see Figure 2 (a)); the third one is when we know the red values in the two adjacent pixels in the same row, namely i is even and j is odd (see Figure 2 (b)); the fourth one is when we know the red values of the pixels adjacent in the corners of the involved pixel, that is i and j are both odd (see Figure 2 (c)). In the second and in the third case we equalize the second derivatives of the red and the green channels previously computed. In the last case we use the computed values of the red channel to determine the weights of a suitable mean. So, we define the initial estimate

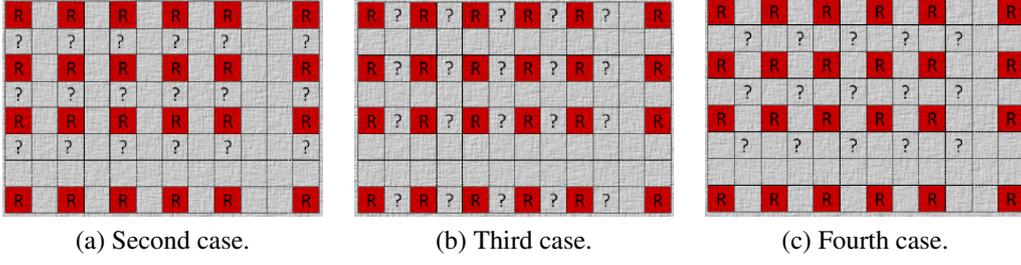


Figure 2: Different cases in the initialization of the red channel.

$\mathbf{r}^{(0)}$ of the red ideal image \mathbf{r} by

$$r_{(i,j)}^{(0)} = \begin{cases} \bar{r}_{(i,j)} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 0, \\ \frac{\bar{r}_{(i-1,j)} + \bar{r}_{(i+1,j)} - (g_{(i-1,j)}^{(0)} - 2\bar{g}_{(i,j)} + g_{(i+1,j)}^{(0)})}{2} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 0, \\ \frac{\bar{r}_{(i,j-1)} + \bar{r}_{(i,j+1)} - (g_{(i,j-1)}^{(0)} - 2\bar{g}_{(i,j)} + g_{(i,j+1)}^{(0)})}{2} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 1, \\ \frac{t_9 r_{(i+1,j)}^{(0)} + t_{10} r_{(i-1,j)}^{(0)} + t_{11} r_{(i,j-1)}^{(0)} + t_{12} r_{(i,j+1)}^{(0)}}{t_9 + t_{10} + t_{11} + t_{12}} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 1. \end{cases}$$

Note that, in the first case, we keep the value which we already have. In the second case, we pose that the finite difference of the second order in the vertical direction of the red channel coincides with that of the green channel, which we already initialized, namely

$$\bar{r}_{(i-1,j)} - 2r_{(i,j)}^{(0)} + \bar{r}_{(i+1,j)} = g_{(i-1,j)}^{(0)} - 2\bar{g}_{(i,j)} + g_{(i+1,j)}^{(0)}. \quad (2)$$

Since we know $\mathbf{g}^{(0)}$, $\bar{\mathbf{g}}$ and $\bar{\mathbf{r}}$, we can deduce the value of $r_{(i,j)}^{(0)}$ from (2).

In the third case, we impose that the finite difference of the second order in the horizontal direction of the red channel coincides with that of the green channel, just already initialized, that is

$$\bar{r}_{(i,j-1)} - 2r_{(i,j)}^{(0)} + \bar{r}_{(i,j+1)} = g_{(i,j-1)}^{(0)} - 2\bar{g}_{(i,j)} + g_{(i,j+1)}^{(0)}. \quad (3)$$

By proceeding analogously as above, we obtain the value of $r_{(i,j)}^{(0)}$ from (3).

In the fourth case, we do a weighted mean of the intensity values taken on the adjacent pixels where the red value has just been computed. The weights $t_9, t_{10}, t_{11}, t_{12}$ of the mean are calculated by using the observed blue and the red channels, this last just initialized in the second and third case. In particular, t_9 is given by $\phi(\tau_9)$, where ϕ is the detection function used in initializing the green channel, and

$$\tau_9 = |r_{(i-1,j)}^{(0)} - r_{(i,j-1)}^{(0)}| + |r_{(i-1,j)}^{(0)} - r_{(i,j+1)}^{(0)}| + |\bar{b}_{(i,j)} - \bar{b}_{(i-2,j)}|.$$

When the differences between the red values on the pixels $(i-1, j)$ and $(i, j-1)$, $(i-1, j)$ and $(i, j+1)$, and between the blue values on the pixels (i, j) and $(i-2, j)$, are sufficiently

small, then we can suppose that there are no edges between the pixels $(i - 1, j)$ and (i, j) . So, in the calculus of the red value on the pixel (i, j) , we have a large weight t_9 in correspondence with the red value in the position $(i - 1, j)$. Analogously as in the previous case, we deduce $t_{10} = \phi(\tau_{10})$, where

$$\tau_{10} = |r_{(i+1,j)}^{(0)} - r_{(i,j-1)}^{(0)}| + |r_{(i+1,j)}^{(0)} - r_{(i,j+1)}^{(0)}| + |\bar{b}_{(i,j)} - \bar{b}_{(i+2,j)}|;$$

$t_{11} = \phi(\tau_{11})$, where

$$\tau_{11} = |r_{(i-1,j)}^{(0)} - r_{(i,j-1)}^{(0)}| + |r_{(i,j-1)}^{(0)} - r_{(i+1,j)}^{(0)}| + |\bar{b}_{(i,j)} - \bar{b}_{(i,j-2)}|;$$

$t_{12} = \phi(\tau_{12})$, where

$$\tau_{12} = |r_{(i,j+1)}^{(0)} - r_{(i-1,j)}^{(0)}| + |r_{(i,j+1)}^{(0)} - r_{(i+1,j)}^{(0)}| + |\bar{b}_{(i,j)} - \bar{b}_{(i,j+2)}|.$$

3.3 The initialization of the blue values

Also in this setting, we distinguish four cases: the first one is given when we know the blue value of a pixel; the second one is when we know the blue values in the two adjacent pixels in the same column, that is i is even and j is odd (see Figure 4 (a)); the third one is when we know the blue values in the two adjacent pixels in the same row, namely i is odd and j is even (see Figure 4 (b)); the fourth one is when we know the blue values of the pixels adjacent in the corners of the involved pixel, that is i and j are both even (see Figure 4 (c)). In the second and third cases we equalize the second derivatives of the blue and the green channels previously calculated. In the last case we use the computed values of the blue channel to determine the weights of a suitable mean.

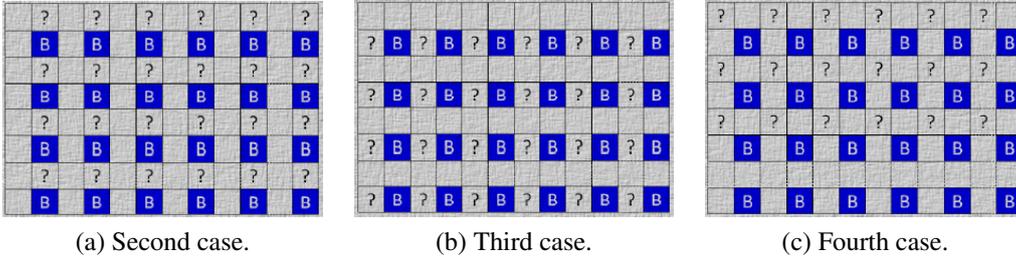


Figure 3: Different cases in the initialization of the blue channel.

Thus, we define the estimate $\mathbf{b}^{(0)}$ of the blue ideal image \mathbf{b} by

$$b_{(i,j)}^{(0)} = \begin{cases} \bar{b}_{(i,j)} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 1; \\ \frac{\bar{b}_{(i-1,j)} + \bar{b}_{(i+1,j)} - (g_{(i-1,j)}^{(0)} - 2\bar{g}_{(i,j)} + g_{(i+1,j)}^{(0)})}{2} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 1; \\ \frac{\bar{b}_{(i,j-1)} + \bar{b}_{(i,j+1)} - (g_{(i,j-1)}^{(0)} - 2\bar{g}_{(i,j)} + g_{(i,j+1)}^{(0)})}{2} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 0; \\ \frac{t_9 b_{(i+1,j)}^{(0)} + t_{10} b_{(i-1,j)}^{(0)} + t_{11} b_{(i,j-1)}^{(0)} + t_{12} b_{(i,j+1)}^{(0)}}{t_9 + t_{10} + t_{11} + t_{12}} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 0. \end{cases}$$

Note that, in the first case, we keep the value which we already have.

In the second case, analogously as before, we impose

$$\bar{b}_{(i-1,j)} - 2b_{(i,j)}^{(0)} + \bar{b}_{(i+1,j)} = g_{(i-1,j)}^{(0)} - 2\bar{g}_{(i,j)} + g_{(i+1,j)}^{(0)}. \quad (4)$$

As we know $g^{(0)}$, \bar{g} and \bar{b} , we derive the value of $b_{(i,j)}^{(0)}$ from (4).

In the third case, similarly as above, we get

$$\bar{b}_{(i,j-1)} - 2b_{(i,j)}^{(0)} + \bar{b}_{(i,j+1)} = g_{(i,j-1)}^{(0)} - 2\bar{g}_{(i,j)} + g_{(i,j+1)}^{(0)}. \quad (5)$$

By arguing as in the previous section, we deduce the value of $b_{(i,j)}^{(0)}$ from (5).

In the fourth case, we do a weighted mean of the intensity values of the adjacent pixels where the blue value has just been computed. The weights $t_9, t_{10}, t_{11}, t_{12}$ of the mean are calculated by using the observed blue channels and the blue channel, just initialized in the second and third case.

Analogously as before, we obtain $t_{13} = \phi(\tau_{13})$, where

$$\tau_{13} = |b_{(i-1,j)}^{(0)} - b_{(i,j-1)}^{(0)}| + |b_{(i-1,j)}^{(0)} - b_{(i,j+1)}^{(0)}| + |\bar{r}_{(i,j)} - \bar{r}_{(i-2,j)}|;$$

$t_{14} = \phi(\tau_{14})$, where

$$\tau_{14} = |b_{(i+1,j)}^{(0)} - b_{(i,j-1)}^{(0)}| + |b_{(i+1,j)}^{(0)} - b_{(i,j+1)}^{(0)}| + |\bar{r}_{(i,j)} - \bar{r}_{(i+2,j)}|;$$

$t_{15} = \phi(\tau_{15})$, where

$$\tau_{15} = |b_{(i-1,j)}^{(0)} - b_{(i,j-1)}^{(0)}| + |b_{(i,j-1)}^{(0)} - b_{(i+1,j)}^{(0)}| + |\bar{r}_{(i,j)} - \bar{r}_{(i,j-2)}|;$$

and finally $t_{16} = \phi(\tau_{16})$, where

$$\tau_{16} = |b_{(i,j+1)}^{(0)} - b_{(i-1,j)}^{(0)}| + |b_{(i,j+1)}^{(0)} - b_{(i+1,j)}^{(0)}| + |\bar{r}_{(i,j)} - \bar{r}_{(i,j+2)}|.$$

4 The iterative phase of the proposed algorithm

A classical filter, often used to solve the demosaicing problem, is the *Freeman filter* (see also [10]). The phase described in this section is a suitable modification of this filter. The Freeman filter performs the initialization phase by means of the bilinear filter, which works as follows. When the value of a certain color of a pixel is not available, such a value is computed by the arithmetic mean of the values of that color, which are assumed in the neighborhood of this pixel, that is the *bilinear estimation* $\tilde{\mathbf{x}} = (\tilde{\mathbf{r}}, \tilde{\mathbf{g}}, \tilde{\mathbf{b}})$ is given as

$$\tilde{g}_{(i,j)} = \begin{cases} \bar{g}_{(i,j)} & \text{if } i \neq_2 j, \\ \frac{\bar{g}_{(i-1,j)} + \bar{g}_{(i+1,j)} + \bar{g}_{(i,j-1)} + \bar{g}_{(i,j+1)}}{4} & \text{otherwise;} \end{cases}$$

$$\tilde{r}_{(i,j)} = \begin{cases} \bar{r}_{(i,j)} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 0, \\ \frac{\bar{r}_{(i,j-1)} + \bar{r}_{(i,j+1)}}{2} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 1, \\ \frac{\bar{r}_{(i-1,j)} + \bar{r}_{(i+1,j)}}{2} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 0, \\ \frac{\bar{r}_{(i-1,j-1)} + \bar{r}_{(i+1,j-1)} + \bar{r}_{(i-1,j+1)} + \bar{r}_{(i+1,j+1)}}{4} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 1; \end{cases}$$

$$\tilde{b}_{(i,j)} = \begin{cases} \bar{b}_{(i,j)} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 1, \\ \frac{\bar{b}_{(i,j-1)} + \bar{b}_{(i,j+1)}}{2} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 0, \\ \frac{\bar{b}_{(i-1,j)} + \bar{b}_{(i+1,j)}}{2} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 1, \\ \frac{\bar{b}_{(i-1,j-1)} + \bar{b}_{(i+1,j-1)} + \bar{b}_{(i-1,j+1)} + \bar{b}_{(i+1,j+1)}}{4} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 0. \end{cases}$$

Moreover, in [10] the following values are defined, by means of the median of the color differences of the channels red-green and blue-green:

$$\begin{aligned} \tilde{r}g_{(i,j)} &= \text{median}\{\tilde{r}_{(k,l)} - \tilde{g}_{(k,l)} : (k,l) \in B_\infty((i,j),t)\}, \\ \tilde{b}g_{(i,j)} &= \text{median}\{\tilde{b}_{(k,l)} - \tilde{g}_{(k,l)} : (k,l) \in B_\infty((i,j),t)\}, \end{aligned}$$

where

$$B_\infty((i,j),t) := \{(k,l) \in \mathbb{N} \times \mathbb{N} : \|(i,j) - (k,l)\|_\infty \leq t\}, \quad (6)$$

with $\|(a,b)\|_\infty = \max\{|a|, |b|\}$. The median turns out to be very useful to correctly preserve the edges which are in the images. Indeed, the median filter is often used to restore images corrupted by salt-and-pepper noise, namely by the noise present only in a few pixels not adjacent each other.

In the Freeman filter it is assumed that the color differences are constant in a suitable subarea. Thus, the *Freeman estimation* $\hat{\mathbf{x}} = (\hat{\mathbf{r}}^T, \hat{\mathbf{g}}^T, \hat{\mathbf{b}})^T$ is defined as follows:

$$\hat{g}_{(i,j)} = \begin{cases} \bar{g}_{(i,j)} & \text{if } i \not\equiv_2 j, \\ \frac{(\tilde{r}_{(i,j)} - \tilde{r}g_{(i,j)}) + (\tilde{b}_{(i,j)} - \tilde{b}g_{(i,j)})}{2} & \text{otherwise;} \end{cases}$$

$$\hat{r}_{(i,j)} = \begin{cases} \bar{r}_{(i,j)} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 0, \\ \bar{g}_{(i,j)} + \tilde{r}g_{(i,j)} & \text{otherwise;} \end{cases}$$

$$\hat{b}_{(i,j)} = \begin{cases} \bar{b}_{(i,j)} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 1, \\ \bar{g}_{(i,j)} + \tilde{b}g_{(i,j)} & \text{otherwise.} \end{cases}$$

In this paper we modify the Freeman filter as follows.

From the initial estimation $\mathbf{x}^{(0)} = (\mathbf{r}^{(0)T}, \mathbf{g}^{(0)T}, \mathbf{b}^{(0)T})^T$ we define the following variables:

$$\begin{aligned} r g_{(i,j)}^{(s)} &= \text{median}\{r_{(k,l)}^{(s)} - g_{(k,l)}^{(s)} : (k,l) \in B_\infty((i,j),t)\}, \\ b g_{(i,j)}^{(s)} &= \text{median}\{b_{(k,l)}^{(s)} - g_{(k,l)}^{(s)} : (k,l) \in B_\infty((i,j),t)\}, \\ r b_{(i,j)}^{(s)} &= \text{median}\{r_{(k,l)}^{(s)} - b_{(k,l)}^{(s)} : (k,l) \in B_\infty((i,j),t)\}, \end{aligned}$$

where $s = 0, 1, \dots$

So, we define the estimates $\mathbf{x}^{(s)} = (\mathbf{r}^{(s)T}, \mathbf{g}^{(s)T}, \mathbf{b}^{(s)T})^T$ for $s = 1, 2, \dots$ as follows:

$$\begin{aligned} g_{(i,j)}^{(s)} &= \begin{cases} \bar{g}_{(i,j)} & \text{if } i \not\equiv_2 j, \\ \frac{(r_{(i,j)}^{(s-1)} - r g_{(i,j)}^{(s-1)}) + (b_{(i,j)}^{(s-1)} - b g_{(i,j)}^{(s-1)})}{2} & \text{otherwise:} \end{cases} \\ r_{(i,j)}^{(s)} &= \begin{cases} \bar{r}_{(i,j)} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 0, \\ \bar{b}_{(i,j)} + r b_{(i,j)}^{(s-1)} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 1, \\ \bar{g}_{(i,j)} + r g_{(i,j)}^{(s-1)} & \text{otherwise;} \end{cases} \\ b_{(i,j)}^{(s)} &= \begin{cases} \bar{b}_{(i,j)} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 1, \\ \bar{r}_{(i,j)} - r b_{(i,j)}^{(s-1)} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 0, \\ \bar{g}_{(i,j)} + b g_{(i,j)}^{(s-1)} & \text{otherwise.} \end{cases} \end{aligned}$$

We pose our final estimate as

$$\check{\mathbf{x}} = \lim_{s \rightarrow +\infty} \mathbf{x}^{(s)}.$$

We saw experimentally that a good approximation is given by $\check{\mathbf{x}} = \mathbf{x}^{(4)}$. We call the technique associated to this estimate as *Local Edge Preserving* (LEP) algorithm.

5 Experimental results and discussion

In this section we present the experimental results obtained from the implementation of the proposed algorithm, which was tested for the Bayer CFA on the set of 24 Kodak sample images [22], of size 512×768 , shown in Figure 4. This dataset represents the typical benchmark images used in the literature to compare the various demosaicing algorithms. These high quality images have been acquired as raw images, in order to minimize the compression.

To define a specific LEP method, we fix to one the radius t of the neighborhood of the median filter in the equation (6), and we experimentally choose the detection function

$\phi : \mathbb{R}_0^+ \rightarrow \mathbb{R}^+$ in (1). In particular, the set of the tested functions consists of

$$\begin{aligned}\phi_1(t) &= \begin{cases} 2 - t & \text{if } 0 \leq t \leq 1, \\ \frac{1}{t} & \text{if } t \geq 1, \end{cases} \\ \phi_2(t) &= \begin{cases} \frac{3-2e}{e-1}t + 2 & \text{if } 0 \leq t \leq 1, \\ \frac{1}{e^t-1} & \text{if } t \geq 1, \end{cases} \\ \phi_3(t) &= \begin{cases} (\log 2 - 2)t + 2 & \text{if } 0 \leq t \leq 1, \\ \frac{1}{\log(t+1)} & \text{if } t \geq 1, \end{cases} \\ \phi_4(t) &= \begin{cases} 2 - t & \text{if } 0 \leq t \leq 1, \\ \frac{1}{t^{13/10}} & \text{if } t \geq 1, \end{cases} \\ \phi_5(t) &= \begin{cases} 2 - t & \text{if } 0 \leq t \leq 1, \\ \frac{1}{t^{7/5}} & \text{if } t \geq 1, \end{cases} \\ \phi_6(t) &= \begin{cases} 2 - t & \text{if } 0 \leq t \leq 1, \\ \frac{1}{t^{3/2}} & \text{if } t \geq 1. \end{cases}\end{aligned}$$

The detection functions ϕ_j , $j = 1, \dots, 6$ are decreasing and continuous. Moreover, we get

$$\phi_j(0) = 2 \text{ and } \lim_{t \rightarrow +\infty} \phi_j(t) = 0.$$

In Table 1 there are the errors of the LEP algorithm in terms of MSE in reconstructing the images of the Kodak set as the detection function varies. The values in bold are related to the best reconstruction of a specific image. In the last line there are the means of the MSE obtained in the reconstruction of the Kodak sample images, as the detection function varies. Note that the best result can be obtained by different detection functions, but, if one takes the means, the minimal error corresponds to detection function ϕ_4 . To evaluate whether the function ϕ_4 is actually the best detection function, we proceed as follows. For each sample image we give five points to the detection function which allows to obtain an estimate with the minimal error; four points to the detection function which obtain the second best minimal error; three points in correspondence with the third minimal error, and so on. In Table 2 there are the results obtained by the all detection functions on the single images, and in the last line there is the global score. Observe that, even in this case, the highest score is obtained by the detection function ϕ_4 .

Successively, we compared the LEP algorithm with other techniques existing in the literature by choosing the detection function ϕ_4 . In particular we compare the LEP method with the original Freeman filter and with other recently published algorithms (see also [5, 9, 12, 16, 20]). Although the proposed algorithm give the best reconstruction of only four images, the total mean of the errors obtained with the LEP algorithm is the smallest one.

6 Conclusions

We investigated the demosaicing problem and proposed a heuristic technique, in order to obtain a very fast algorithm. In particular, we proposed an algorithm consisting of three steps. In the first one, the green channel was updated by means of an edge-directed and weighted average technique. In the second one, the red and blue channels were updated,



(a) Image 01.



(b) Image 02.



(c) Image 03.



(d) Image 04.



(e) Image 05.



(f) Image 06.



(g) Image 07.



(h) Image 08.



(i) Image 09.



(j) Image 10.



(k) Image 11.



(l) Image 12.



(m) Image 13.



(n) Image 14.



(o) Image 15.



(p) Image 16.



(q) Image 17.



(r) Image 18.



(s) Image 19.



(t) Image 20.



(u) Image 21.



(v) Image 22.



(w) Image 23.

Figure 4: Kodak image set.

Image	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6
01	10.02	16.86	13.68	9.77	9.75	9.75
02	6.96	8.66	7.63	6.90	6.91	6.92
03	4.22	6.01	4.95	4.14	4.13	4.12
04	6.16	9.15	6.66	6.17	6.18	6.21
05	13.41	22.46	15.05	13.36	13.39	13.44
06	9.29	13.12	11.76	9.10	9.09	9.08
07	5.05	8.10	5.77	5.04	5.05	5.07
08	20.63	26.39	27.39	19.99	19.87	19.83
09	4.60	7.28	5.68	4.63	4.66	4.69
10	4.82	6.94	5.82	4.79	4.80	4.81
11	7.79	11.17	8.99	7.70	7.70	7.71
12	3.88	5.81	5.41	3.83	3.84	3.85
13	19.05	29.38	20.47	19.11	19.19	19.27
14	15.96	22.12	17.59	15.79	15.77	15.77
15	8.77	11.77	10.29	8.68	8.69	8.71
16	4.26	5.36	5.32	4.11	4.08	4.06
17	5.50	7.83	6.00	5.52	5.54	5.57
18	14.88	21.30	15.36	15.00	15.04	15.09
19	8.68	11.36	11.24	8.48	8.47	8.46
20	6.71	8.24	8.75	6.43	6.39	6.36
21	8.14	12.32	9.41	8.06	8.07	8.08
22	12.12	16.09	12.93	12.09	12.11	12.13
23	3.82	6.17	4.02	3.85	3.87	3.89
mean	8.9002	12.7781	10.4428	8.8074	8.8080	8.8203

Table 1: MSE of the LEP algorithm on the Kodak set as the detection function varies.

by using also the constraint of equality of the second derivatives in the various channels. In the third step, we proposed an iterative algorithm, assuming the constant color difference. Moreover, similarly as in the Freeman technique, in this phase we employed a median filter. We fixed a maximum number of iterative steps as four, in order to obtain low computational costs. We called our algorithm *Local Edge Preserving* (LEP). The experimental results showed that the mean of the errors which we have with the LEP algorithm is the lowest of those obtained with some recent proposed methods.

Conflict of interest

We have no conflict of interest to declare.

Acknowledgements

The first author was partially supported by the G. N. A. M. P. A (the Italian National Group of Mathematical Analysis, Probability and Applications).

References

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

Image	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6
01	2	0	1	3	4	5
02	2	0	1	4	5	3
03	2	0	1	3	4	5
04	5	0	1	4	3	2
05	3	0	1	5	4	2
06	2	0	1	3	4	5
07	4	0	1	5	3	2
08	2	1	0	4	3	5
09	5	0	1	4	3	2
10	2	0	1	5	4	3
11	2	0	1	4	5	3
12	2	0	1	5	4	3
13	5	0	1	4	3	2
14	2	0	1	3	5	4
15	2	0	1	5	4	3
16	2	0	1	3	4	5
17	5	0	1	4	3	2
18	5	0	1	4	3	2
19	2	0	1	3	4	5
20	2	1	0	3	4	5
21	2	0	1	5	4	3
22	3	0	1	5	4	2
23	5	0	1	4	3	2
total	68	2	21	92	87	75

Table 2: Points of the LEP algorithm on the Kodak set as the detection function varies.

- [2] J.E. Adams, and J.F. Hamilton, Design of practical color filter array interpolation algorithms for digital cameras, in Proceeding of SPIE, vol. 3028, 1997 , pp. 117–125.
- [3] C. Bai, J. Li, Z. Lin, J. Yu, and Y.-W. Chen, "Penrose demosaicking", *IEEE Transaction on Image Processing*, vol. 24, pp. 1672–1684, 2015.
- [4] B. E. Bayer, "Color imaging array", U.S. Patent 3971065, 1976.
- [5] A. Buades, B. Coll, J. M. Morel, and C. Sbert, "Self similarity driven color demosaicking", *IEEE Transactions on Image Processing*, vol. 18, pp. 1192–1202, 2009.
- [6] K. Chung, and Y. Chan, "Color demosaicing using variance of color differences", *IEEE Transactions on Image Processing*, vol. 15, pp. 2944–2955, 2006.
- [7] K. Chung, and Y. Chan, "Low complexity color demosaicing algorithm based on integrated gradients", *J. Electron. Imaging*, vol. 19, pp. 1–15, 2010.
- [8] L. Condat, "A simple, fast and efficient approach to denoisaicking: joint demosaicking and denoising", in Image Processing (ICIP), 2010 17th IEEE International Conference on, 2010, pp. 905–908.
- [9] S. Ferradans, M. Bertalmío, and V. Caselles, "Geometry based demosaicking", *IEEE Transactions on Image Processing*, vol. 18, pp. 665–670, 2009.
- [10] W. T. Freeman, "Median filter for reconstructing missing color samples", U.S. Patent 4, 774, 565, 1988.

Image	[16]	[5]	[10]	[9]	[20]	[12]	LEP
01	19.93	35.93	53.34	15.43	17.89	11.04	9.77
02	7.63	36.77	11.69	6.19	14.81	7.81	6.91
03	4.62	37.54	8.57	3.55	12.35	4.64	4.13
04	8.50	37.02	10.04	4.97	13.88	6.59	6.17
05	17.64	35.77	39.02	12.04	18.68	11.49	13.36
06	11.52	36.68	37.33	8.84	15.05	10.69	9.10
07	5.36	37.37	10.05	3.64	12.87	4.54	5.04
08	27.32	34.87	96.56	22.99	22.78	19.99	19.99
09	5.09	37.71	13.06	4.08	11.76	4.30	4.63
10	5.49	37.54	12.29	4.08	12.32	4.34	4.79
11	11.71	36.68	24.57	8.33	14.98	7.59	7.70
12	4.32	37.89	12.90	3.39	11.49	4.80	3.83
13	47.37	33.92	77.41	35.93	28.09	24.38	19.11
14	18.47	35.36	26.53	11.42	20.39	16.42	15.79
15	10.34	36.52	16.74	8.44	15.65	8.72	8.68
16	4.78	37.80	16.61	3.62	11.57	4.23	4.11
17	7.79	41.45	13.28	5.35	5.10	4.90	5.52
18	19.79	36.26	34.73	16.92	16.54	13.24	15.00
19	9.39	40.50	34.55	7.25	6.26	6.83	8.48
20	7.82	32.77	22.11	6.14	36.94	5.80	6.43
21	14.47	36.77	29.09	10.83	14.77	8.37	8.06
22	14.81	37.63	21.57	9.49	12.15	10.33	12.09
23	4.25	39.76	6.97	3.08	7.44	4.07	3.85
mean	12.5396	36.9787	26.9135	9.3913	15.3848	8.9175	8.8074

Table 3: MSE estimates on the Kodak set images of the algorithms in [5, 9, 10, 12, 16, 20] and of the LEP method.

- [11] I. Gerace, F. Martinelli, and A. Tonazzini, "Demosaicing of noisy color images through edge-preserving regularization", in Proceeding of 2014 International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM), 2014, pp. 1–5.
- [12] B. K. Gunturk, Y. Altunbasak, and R.M. Mersereau, "Color plane interpolation using alternating projections", *IEEE Transactions on Image Processing*, vol. 11, pp. 997–1013, 2002.
- [13] J. Hadamard, *Lectures on Cauchy's Problem in Linear Partial Differential Equations*. Yale: New Haven, University Press, 1923.
- [14] J. F. Hamilton, and J. T. Compton, "Processing color and panchromatic pixels", U.S. Patent 2007 0024879, 2007.
- [15] R. H. Hibbard, "Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients", U.S. Patent 5 382 976, 1995.
- [16] K. Hirakawa, and T. W. Parks, "Adaptive homogeneity directed demosaicing algorithm", *IEEE Transactions on Image Processing*, vol. 14, pp. 360–369, 2005.
- [17] K. Hirakawa, and T. W. Parks, "Joint demosaicing and denoising", *IEEE Transactions on Image Processing* vol. 15, pp. 2146–2157, 2006.
- [18] K. Hirakawa, and P. Wolfe, "Spatio-spectral color filter array design for enhanced image fidelity", in Image Processing (ICIP), 2007 IEEE International Conference on, 2007, pp. 81–84.

- [19] R. Kakarala, and Z. Baharav, "Adaptive demosaicking with the principal vector method", *IEEE Transaction on Consumer Electron.*, vol. 48, pp. 932–937, 2002.
- [20] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Beyond color difference: Residual interpolation for color image demosaicking," *IEEE Transaction on Image Processing*, vol. 25, pp. 1288–1300, 2016.
- [21] R. Kimmel, "Demosaicing: image reconstruction from CCD samples", *IEEE Transaction on Image Processing*, vol. 8, pp. 1221–1228, 1999.
- [22] Kodak Lossless True Color Image Suiter, web page <http://r0k.us/graphics/kodak/>
- [23] C. A. Laroche, and M.A. Prescott, "Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients", U. S. Patent 5, 373, 32, 1994.
- [24] W. Lu, and Y.-P. Tan, "Color filter array demosaicing: New method and performance measures", *IEEE Trans. Image Processing*, vol. 12, pp. 1194–1210, 2003.
- [25] J. Mairal, M. Elad, and G. Sapiro, Sparse representation for color image restoration, *IEEE Transactions on Image Processing* vol. 17, pp. 53–69, 2008.
- [26] D. Menon, and G. Calvagno, "Demosaicing based on wavelet analysis of the luminance component", in Image Processing (ICIP), 2007 14th IEEE International Conference on, 2007, pp. 181–184.
- [27] D. Menon, and G. Calvagno, "Regularization approaches to demosaicking", *IEEE Transactions on Image Processing* vol. 18, pp. 2209–2220, 2009.
- [28] D. Menon, and G. Calvagno, "Joint demosaicking and denoising with space-varying filters", in Image Processing (ICIP), 2009 16th IEEE International Conference on, 2009, pp. 477–480.
- [29] D. Menon, and G. Calvagno, "Color image demosaicing: an overview", *Signal Processing: Image Communication* vol. 26, pp. 518–533, 2011.
- [30] S.-C. Pei, and I.-K. Tam, "Effective color interpolation in CCD color filter arrays using signal correlation", *IEEE Trans. Circuits Syst. Video Technol.* vol. 13, pp. 503–513, 2003.
- [31] T. Saito, and T. Komatsu, "Demosaicing approach based on extended color total-variation regularization", in Image Processing (ICIP), 2008 15th IEEE International Conference on, 2008, pp. 885–888.
- [32] B. Tao, I. Tastl, T. Cooper, M. Blasgen, and E. Edwards, "Demosaicing using human visual properties and wavelet interpolation filtering", in Proceeding of Color Imaging Conference: Color Science, Systems, Applications, 1999, pp. 252–256.
- [33] P.-S. Tsai, T. Acharya, and A.K. Ray, "Adaptive fuzzy color interpolation", *J. Electron. Imaging*, vol. 11, pp. 293–305, 2002.

- [34] J. A. Weldy, "Optimized design for a single-sensor color electronic camera system", in *Proceeding of SPIE*, vol. 1071, 1988, pp. 300–307.
- [35] X. Wu, W. K. Choi, and P. Bao, "Color restoration from digital camera data by pattern matching", in *Proceeding of SPIE*, vol. 3018, 1997, pp. 12–17.
- [36] L. Zhang, X. Wu, and D. Zhang, "Color reproduction from noisy CFA data of single sensor digital cameras", *IEEE Transactions on Image Processing*, vol. 16, pp. 2184–2197, 2007.