Edge detection from Bayer CFA images

Arezki ABERKANE[1], Olivier LOSSON[2] and Ludovic MACAIRE[3]

Laboratoire CRIStAL, Université Lille1 - Sciences et Technologies,

Cité scientifique - Bâtiment P2, 59650 Villeneuve d'Ascq Cedex, France

[1]arezki.aberkane@ed.univ-lille1.fr, [2]olivier.losson@univ-lille1.fr, [3]ludovic.macaire@univ-lille1.fr

Abstract

I. Introduction

There are two major families of digital colour cameras: cameras with three sensors where each captures one of the three primary colours (red, green, or blue) and those with a single sensor. Although the digital cameras with three sensors are very effective and give excellent images, this technology is very onerous and cumbersome, which is prohibitive for most consumers. For these reasons, most digital colour cameras nowadays embed only a single sensor. Such cameras are fitted with a Colour Filter Array (CFA) and deliver CFA images, in which each pixel is characterized by only one out of the three colour components (red, green, or blue), and that must be interpolated to estimate the full colour images. This process is known as demosaicing (or demosaicking).

Most demosaicing schemes are dedicated to the widespread Bayer CFA that is presented in section II together with the demosaicing problem. Review papers about the many demosaicing schemes proposed in the literature were conducted by Gunturk et al. [14, 24] and most recently by Menon and Calvagno [30]. Since demosaicing is not our main concern and we precisely aim at avoiding it, we only give a quick insight of the main strategies that have been used to this aim. We especially focus on the most recent and efficient state-of-the-art algorithms, as well as on issues related to CFA image noise.

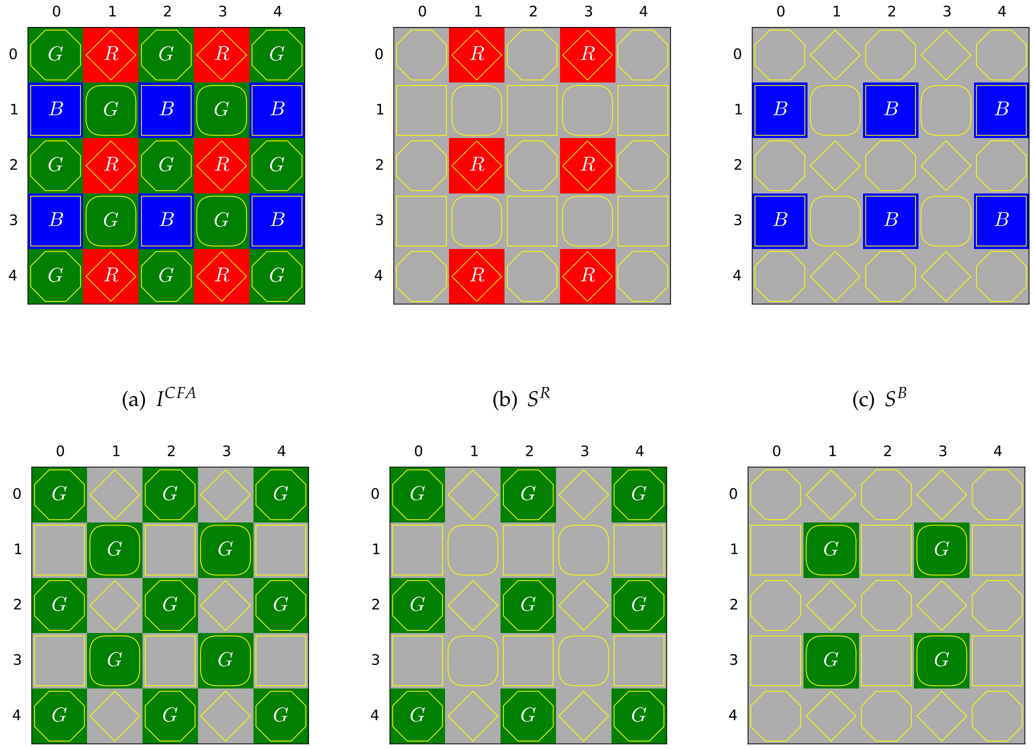
Demosaicing methods are generally designed to produce “perceptually satisfying” images, but do not consider how resultant artefacts would affect a subsequent analysis of these images. In a previous work [27] we show that demosaicing is detrimental to low-level image analysis because it often fails to reconstruct the high-frequency information correctly. Focusing on the various demosaicing artefacts, we propose specific measurements to quantify how each of them specifically occurs and affects edge detection in the demosaiced image. Other studies confirm the latter and show that demosaicing alters the texture representation quality in particular, and that the CFA image can be used effectively for colour texture classification [26, 28]. However, very few works similarly try to use the CFA image directly for low-level image analysis. Chen et al. [6] investigate the feasibility of edge detection by a dedicated Gaussian smoothing and Laplacian-based edge kernels which can be applied to a CFA image. Despite that this method is visually highly noise-sensitive and has not been objectively evaluated, its encouraging results lead us to investigate edge detection from CFA images to avoid artefacts generated by the demosaicing process.

To perform edge detection directly on the CFA image, we adapt Di Zenzo's vector gradient [38] to the CFA image. The colour gradient is based on a tensor of the first derivatives of colour component images. Our main contribution is to adapt these derivative estimation to the CFA image where two colour component levels are missing at each pixel, as presented in section III.

II. CFA demosaicing

II.1. Demosaicing problem formulation

Most of single-sensor cameras are fitted with a Bayer CFA, which is exclusively considered in this paper. Such cameras provide a raw image denoted as I^{CFA} and shown in Fig. 1(a). The set S of all image pixels can be divided into $S = S^R \cup S^G \cup S^B$, where S^k denotes the set of pixels where the colour component k is available in I^{CFA} (see Figs. 1(b)–1(d)). Both S^R and S^B contain pixels arranged in a regular lattice (a quarter of all pixels), and S^G contains (half of all) pixels arranged in a quincunx lattice. It can be further divided into $S^G = S^{G,R} \cup S^{G,B}$, where $S^{G,k}$ ($k \in \{R, B\}$) is the subset of S^G pixels whose horizontal neighbours belong to S^k (see Figs. 1(e) and 1(f)).



(a) I^{CFA} (b) S^R (c) S^B

(d) S^G (e) $S^{G,R}$ (f) $S^{G,B}$

Figure 1: *Bayer CFA image and component-wise pixel subsets. The notations R, G, and B express that the level of the respective colour component is available at that location.*

To determine the colour of each pixel P in the estimated colour image $\hat{\mathbf{I}}$, the demosaicing process generally retains the colour component available at the same location in I^{CFA}, and estimates the missing other two components:

$$\hat{\mathbf{I}}(P) = \begin{cases} \left(I^{CFA}(P), \hat{I}^G(P), \hat{I}^B(P)\right)^T & \text{if } P \in S^R, \\ \left(\hat{I}^R(P), I^{CFA}(P), \hat{I}^B(P)\right)^T & \text{if } P \in S^G, \\ \left(\hat{I}^R(P), \hat{I}^G(P), I^{CFA}(P)\right)^T & \text{if } P \in S^B. \end{cases} \tag{1}$$

Each triplet of colour component levels in Eq. (1) represents an estimated colour. Out of the three components of $\hat{\mathbf{I}}(P)$, the one denoted by $I^{CFA}(P)$ is available at P in I^{CFA}, and the other two among $\hat{I}^R(P)$, $\hat{I}^G(P)$ and $\hat{I}^B(P)$ are estimated by demosaicing because they are unavailable.

II.2. Demosaicing schemes in literature

Bilinear interpolation [34] is the simplest interpolation algorithm. It estimates the two colour components missing at each pixel by averaging the values of same component available at its neighbours. This method gives rise to severe artefacts in image areas with high spatial frequencies [5] because its basic intra-channel interpolation both ignores inter-channel correlation and lacks any edge-sensing mechanism.

As shown in [14], all colour channels of natural images largely share a common high-frequency content (i.e., have similar textures and edge locations). Many demosaicing schemes rely on this strong spectral correlation and use the colour channel difference (CD) domains ($R-G$ and $B-G$) where the reduced high-frequency energy simplifies the interpolation process. Moreover, the second key principle of spatial correlation states to avoid interpolating missing components at a given pixel using neighbour levels that do not belong to the same homogeneous region. This yields to determine the direction according to which the interpolation should be performed to ensure an artefact-free edge restoration. This strategy has grounded many algorithms [30], from a fairly old patent by Hamilton and Adams [15] and similar hard-decision schemes that try to select the optimal interpolation direction, to the most sophisticated soft-decision schemes that adaptively combine several interpolation results from different directions. Pekkucuksen and Altunbasak [32] recently proposed a very efficient approach of this kind, that weights each directional interpolation result by the inverse of the squared CD gradient summed up over a local window.

Recently have emerged another family of demosaicing schemes that rely on residual interpolation (RI) [19]. A residual is defined as the difference between the acquired ("genuine") value and a tentatively estimated value. Thanks to the edge-preserving guided filtering (GF) [16] that provides accurate estimates, residuals can be made small enough for the residual domain to be much smoother than the CD domain, hence more appropriate to efficient demosaicing [37]. Kiku et al. [19] replace the linear (horizontal or vertical) CD interpolation by an RI in the gradient-based threshold-free algorithm of Pekkucuksen and Altunbasak [32], and estimate colour planes with each other as guides. For instance, to get $\hat{I}^R(P), P \in S^G$, the authors i) generate a fully-defined G channel by linear CD interpolation, ii) use it as a guide image to upsample the pixels in S^R by GF, iii) subtract these tentative estimates to the genuine values at the pixels in S^R to get the residuals, and iv) interpolate the residuals bilinearly. Ye and Ma [37] iterate the RI process to refine the estimated G channel. At each step, the horizontally- and vertically-estimated G-component images are weighted by the inverse of the squared residual gradient. Kiku et al. [20, 21] also improve their original algorithm by minimizing the Laplacian energies of the residuals, instead of the sum of their squared differences as in GF. Wang and Jeon [36] propose a hybrid approach that first uses the CD-based weighted interpolation [7] over eight directions to estimate $\hat{G}$, then the GF with $\hat{G}$ as guidance and RI to estimate $\hat{R}$ and $\hat{B}$.

The performance of RI-based strategies on the Kodak dataset, that is widely used as a benchmark in the demosaicing literature, is a little beneath that of state-of-the-art CD-based ones. But it is the opposite on the IMAX dataset, for which they even outperform methods that rely on compressive sensing and require to learn a sparsifying dictionary [29, 31, 33]. This is because RI-based strategies less rely on spectral correlation and are less sensitive to sharp colour transitions that characterize IMAX images [12].

Let us note at last that other interesting approaches have been proposed in the literature. For example, frequency-based approaches [3] rely on the Fourier decomposition of a Bayer CFA image that can be represented as a combination of one luminance and two chrominance signals. Despite that these signals are rather well localized in the frequency domain, designing an appropriate frequency selection to estimate them (and then the demosaiced image) is still a challenge [11, 23]. Wavelet-based approaches (e.g., [25, 22]) and non-local image self-similarity approaches (e.g., [12]) perform well when inter-channel correlation is low or local geometry is ambiguous, but at a rather high computational cost.

II.3. CFA image denoising

The demosaiced image quality both depends on the demosaicing scheme itself, but also on the noise inherent to the acquisition process. Post-processing the demosaiced image by a denoising scheme is unsatisfying because this generates many noise-caused colour artefacts [39]. Most demosaicing schemes have indeed been designed under the assumption of noise-free data and introduce a spatial correlation in the noise characteristics, which makes it very hard to remove.

Since both demosaicing and denoising rely on an estimation of a sample from its neighbours, some schemes propose to perform the two tasks jointly [8, 13, 18]. Although this approach improves the final image quality and provides relatively fast reconstructions, it prevents an independent design of the denoising and demosaicing algorithms. Several works propose instead to denoise the CFA image before demosaicing it, with very interesting results too.

Classical denoising algorithms cannot be applied on the CFA image due to its mosaic structure. Zhang et al. [39] propose a PCA-based denoising method dedicated to the CFA image that uses a covariance matrix estimated from similar blocks in a neighbourhood. The PCA property of optimal dimensionality reduction guarantees a successful noise attenuation (by resetting the least significant components) and a good signal recovery (by further using a linear minimum mean squared-error estimation of the remaining components). To avoid phantom artefacts in smooth areas, the authors also propose to decompose the noisy CFA image into low- and high-pass images and to perform the PCA on the sole latter since it contains most of the noise. Danielyan et al. [9] use the Block Matching 3D (BM3D) filter, that exploits non-local similarity of small image patches (blocks). The authors slightly modify the original greyscale algorithm so that all blocks in a group have the same CFA pattern. Akiyama et al. [2] consider overlapping 2×2 pixel blocks in the raw CFA data to form four pseudo four-channel images, transform each channel via PCA as in [39], and use the greyscale BM3D filter to denoise it in the principal component domain.

The four denoised four-channel images are then rearranged pixel-wise back as CFA blocks, and averaged to get the denoised CFA raw data.

The BM3D filter was originally designed for additive white Gaussian noise, but the authors in [9] successfully apply it to CFA images corrupted by a signal-dependent noise. This may be a more realistic noise model for digital raw data [35], but the channel-dependent additive noise model (i.e., channel-wise signal-independent noises) used in [39] and [2] is a reasonable one for the white-balanced gamma-corrected raw signal [17].

III. Colour and CFA gradients

In this section, we first briefly recall how Di Zenzo [38] propose to compute the vector gradient. Then, we adapt it to CFA images to perform edge detection directly on such images.

III.1. Di Zenzo's colour gradient

To compute the gradient on a colour image $\mathbf{I} = \left(I^R, I^G, I^B\right)$, Di Zenzo [38] estimates the first partial derivatives of each colour component $k \in \{R, G, B\}$ according to x and y, below shortly denoted as $I_x^k \doteq \frac{\partial I^k}{\partial x}$ and $I_y^k \doteq \frac{\partial I^k}{\partial y}$ and regrouped into two vectors $\mathbf{I}_x = \left(I_x^R, I_x^G, I_x^B\right)^T$ and $\mathbf{I}_y = \left(I_y^R, I_y^G, I_y^B\right)^T$ called partial derivatives of the colour image.

They are then used to compute the gradient direction θ^* and magnitude $||\nabla\mathbf{I}||$ at each pixel P

(omitted here to alleviate equations) by finding the value of θ that maximizes the first fundamental form:

$$d\mathbf{I}^2(\theta) = a\cos^2\theta + 2b\cos\theta\sin\theta + c\sin^2\theta, \tag{2}$$

where $a = \mathbf{I}_x \cdot \mathbf{I}_x = \sum_{k=R,G,B}(I_x^k)^2$, $b = \mathbf{I}_x \cdot \mathbf{I}_y = \sum_{k=R,G,B} I_x^k \cdot I_y^k$, $c = \mathbf{I}_y \cdot \mathbf{I}_y = \sum_{k=R,G,B}(I_y^k)^2$. This compass approach provides the colour gradient as: $\theta^* = \underset{\theta\in[-\pi,\pi]}{\operatorname{argmax}}\, d\mathbf{I}^2(\theta)$, $||\nabla\mathbf{I}||^2 = \left|d\mathbf{I}^2(\theta^*)\right|$. Equivalently, maximal variations of $\mathbf{I}$ are given by the closed-form solution $\theta^* = \frac{1}{2}\cdot\arctan\left(2b/(a-c)\right)$, $||\nabla\mathbf{I}||^2 = \frac{1}{2}\cdot\left(a + c + \sqrt{(a-c)^2 + 4b^2}\right)$.

III.2. Estimation partial derivative

a Simple partial derivative

Assuming that I^k is continuous and derivable, these derivatives may for instance be approximated at each pixel $P(x,y)$ as simply as:

$$\begin{aligned} I_x^k(x,y) &\approx \tfrac{1}{2}\cdot\left(I^k(x+1,y) - I^k(x-1,y)\right), \\ I_y^k(x,y) &\approx \tfrac{1}{2}\cdot\left(I^k(x,y-1) - I^k(x,y+1)\right). \end{aligned} \tag{3}$$

$k = \{R,G,B\}$

b Deriche partial derivative

The Deriche approach [10] is often preferred to Canny filter [4] despite that exactly meets the same criterion of Canny, but which has an infinite impulse response (RIII filter). Firstly, it is advantageous to introduce the spatial distance d separating the considered pixel from its horizontal or vertical nearest neighbour. To estimate the partial derivatives of each colour component $k \in \{R, G, B\}$ with Deriche filter is set to $d = 1$, because the colour component image I^k is fully defined. To compute I_x^k, a smoothing is first applied in the vertical direction (according to y) from top to bottom (see equation (4)) and from bottom to top (see equation (5)):

- Smoothing according to y

$$\bar{I}_y^{k^-}(x,y) = a_1 \cdot I^k(x,y) + a_2 \cdot I^k(x,y-d) + b_1 \cdot \bar{I}_y^{k^+}(x,y-d) + b_2 \cdot \bar{I}_y^{k^+}(x,y-2d) \tag{4}$$

$$\bar{I}_y^{k^+}(x,y) = a_3 \cdot I^k(x,y+d) + a_4 \cdot I^k(x,y+2d) + b_1 \cdot \bar{I}_y^{k^+}(x,y+d) + b_2 \cdot \bar{I}_y^{k^+}(x,y+2d) \tag{5}$$

The vertically smoothed image $\bar{I}_y^k$ by the Deriche filter is given by:

$$\bar{I}_y^k(x,y) = \bar{I}_y^{k^-}(x,y) + \bar{I}_y^{k^+}(x,y) \tag{6}$$

$$\textit{Where}\ \begin{cases} a_1 = m \\ a_2 = m \cdot (\alpha - 1) \cdot \mathrm{e}^{-\alpha} \\ a_3 = m \cdot (\alpha + 1) \cdot \mathrm{e}^{-\alpha} \\ a_4 = -m \cdot \mathrm{e}^{-2\cdot\alpha} \end{cases} \quad \text{and} \quad \begin{cases} b_1 = 2 \cdot \mathrm{e}^{-\alpha} \\ b_2 = -\,\mathrm{e}^{-2\cdot\alpha} \\ m = \frac{(1-\mathrm{e}^{-\alpha})^2}{1+2\cdot\alpha\cdot\mathrm{e}^{-\alpha} - \mathrm{e}^{-2\cdot\alpha}} \end{cases} \tag{7}$$

Now, for compute partial derivative according to x, $\bar{I}_y^k$ is scanned in the horizontal direction from left to right (see equation (8)) and from right to left (see equation (9)):

- Derivative according to x

$$I_x^{k^-}(x,y) = a_5 \cdot \bar{I}_y^k(x,y) + a_6 \cdot \bar{I}_y^k(x-d,y) + b_1 \cdot I_x^{k^-}(x-d,y) + b_2 \cdot I_x^{k^-}(x-2d,y) \tag{8}$$

$$I_x^{k^+}(x,y) = a_7 \cdot \bar{I}_y^k(x+d,y) + a_8 \cdot \bar{I}_y^k(x+2d,y) + b_1 \cdot I_x^{k^+}(x+d,y) + b_2 \cdot I_x^{k^+}(x+2d,y) \tag{9}$$

The partial derivative according y by the Deriche filter is given by:

$$I_x^k(x,y) = I_x^{k^+}(x,y) + I_x^{k^-}(x,y) \tag{10}$$

The parameters that changes compared to the equation (7) are: $a_5 = 0$, $a_6 = 1$, $a_7 = -1$ and $a_8 = 0$.

The partial derivatives according to y I^k are computed in the same way.

III.3. Proposed CFA image-based vector gradient

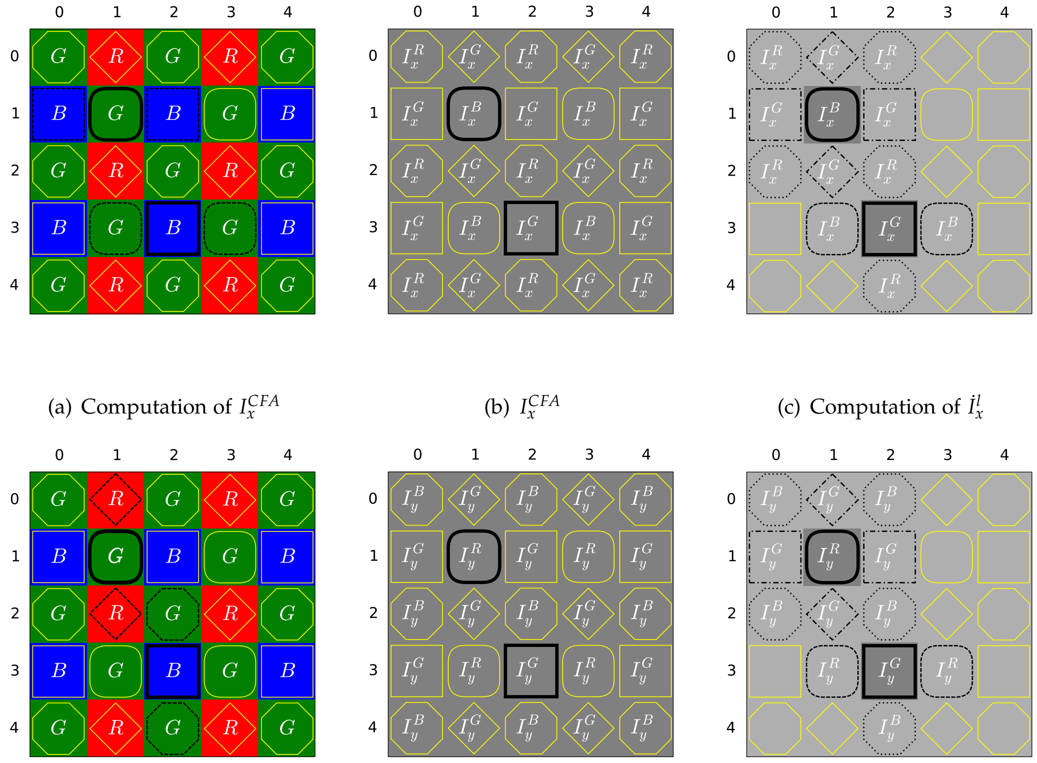
We follow Di Zenzo's approach designed for a fully-defined (colour) image, and adapt it to a CFA image. The key issue here is to estimate the partial derivatives of $\mathbf{I}$ from I^{CFA}, that we denote as $\dot{\mathbf{I}}_x$ and $\dot{\mathbf{I}}_y$.

a Simple partial derivative

As a starting point, we compute the first partial derivatives I_x^{CFA} and I_y^{CFA} of a CFA image at each pixel $P(x,y)$ using the same simple digital approximations as in Eq.(3):

$$\begin{aligned} I_x^{CFA}(x,y) &\approx \tfrac{1}{2}\cdot\left(I^{CFA}(x+1,y) - I^{CFA}(x-1,y)\right), \\ I_y^{CFA}(x,y) &\approx \tfrac{1}{2}\cdot\left(I^{CFA}(x,y-1) - I^{CFA}(x,y+1)\right). \end{aligned} \tag{11}$$

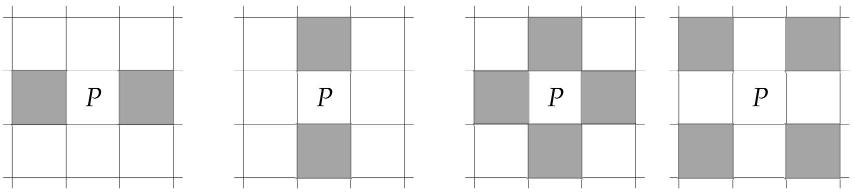
These derivatives are directly obtained from I^{CFA} considered here as a simple grey scale image, and can be viewed as the result of a spatial multiplexing of the components of the same derivatives of the reference image $\mathbf{I}$ spectrally sub-sampled according to the CFA mosaic. Because the two horizontal (or vertical) neighbours of each pixel P belong to the same CFA subset S^k, their



(a) Computation of I_x^{CFA} (b) I_x^{CFA} (c) Computation of $\dot{I}_x^l$

(d) Computation of I_y^{CFA} (e) I_y^{CFA} (f) Computation of $\dot{I}_y^l$

Figure 2: *First partial derivatives of a CFA image (see Eqs.* (12)– (16)*), with two pixels as examples (solid bold lines). Dashed neighbours in Figs. (a) and (d) are used to compute I_x^{CFA} and I_y^{CFA}. Dashed, dotted, and dash-and-dotted neighbours in Figs. (c) and (f) are used to compute the missing derivative components.*



(a) $N^x(P)$ (b) $N^y(P)$ (c) $N^+(P)$ (d) $N^\times(P)$

Figure 3: *Neighbourhoods of a pixel P used to interpolate the missing derivative components from a CFA image.*

level difference matches with the horizontal (or vertical) derivative I_x^k (or I_y^k) of the colour component k (see Fig. 2). For instance, the horizontal neighbours of the pixel $P(2,3) \in S^B$ in Fig. 2(a) belong to S^G and their level difference equals $2 \cdot I_x^G$ as shown in Fig. 2(b). Its vertical neighbours also belong to S^G (see Fig. 2(d)) and their level difference equals $2 \cdot I_y^G$ (see Fig. 2(e)). The pixels in S^G have to be considered separately according to whether they belong belong to $S^{G,R}$ or $S^{G,B}$. For instance, the horizontal (resp., vertical) neighbours of the pixel $P(1,1) \in S^{G,B}$ belong to S^B (resp., S^R) and their level difference equals $2 \cdot I_x^B$ (resp., $2 \cdot I_y^R$). More generally, I_x^{CFA} (resp., I_y^{CFA}) matches with a component of $\mathbf{I}_x$ (resp., $\mathbf{I}_y$) at each pixel P according to:

$$I_x^{CFA}(P) = \begin{cases} I_x^G(P) & \text{if } P \in S^R \cup S^B, \\ I_x^R(P) & \text{if } P \in S^{G,R}, \\ I_x^B(P) & \text{if } P \in S^{G,B}, \end{cases} \quad \text{and} \quad I_y^{CFA}(P) = \begin{cases} I_y^G(P) & \text{if } P \in S^R \cup S^B, \\ I_y^B(P) & \text{if } P \in S^{G,R}, \\ I_y^R(P) & \text{if } P \in S^{G,B}. \end{cases} \tag{12}$$

b CFA Deriche partial derivative

In section III.2.b, to estimate the Deriche partial derivative of a colour image $\mathbf{I}$, we calculated these partial derivatives in each colour component I^k, $k \in \{R, G, B\}$. Similarly, in the CFA images, the estimation of the first partial derivatives should be at the same colour component. Each colour component in the CFA image is not available at each pixel $P(x, y)$, according to figures 1(b), 1(c) and 1(d), the nearest neighbour of the pixel $P(x, y) \in S^k$, $k \in \{R, G, B\}$, that also belongs to S^k, is located at a spatial distance $d = 2$ according to horizontal and vertical direction. So, to estimate the Deriche partial derivative of a CFA image at each pixel $P(x, y)$, we proceed in the same way as in section III.2.b by setting d to 2. For example, the partial derivative I_x^{CFAD} by the Deriche filter is $I_x^{CFAD}(x, y) = I_x^{CFAD^-}(x, y) + I_x^{CFAD^+}(x, y)$ where $I_x^{CFAD^-}(x, y)$ and $I_x^{CFAD^+}(x, y)$ are calculated in the same way as equations (8) and (9) but with $d = 2$.

More generally, I_x^{CFAD} (resp., I_y^{CFAD}) matches with a component of $\mathbf{I}_x$ (resp., $\mathbf{I}_y$) at each pixel P according to:

$$I_x^k(P) = I_x^{CFAD}(P) \quad \text{if } P \in S^k, \quad \text{and} \quad I_y^k(P) = I_y^{CFAD}(P) \quad \text{if } P \in S^k, \tag{13}$$

With $k \in \{R, G, B\}$.

c Fully defined partial derivative

To get the fully-defined partial derivatives $\dot{\mathbf{I}}_x$ and $\dot{\mathbf{I}}_y$ of the colour image $\mathbf{I}$ from I^{CFA}, we need to estimate the two derivative components missing at each pixel P in I_x^{CFA} and I_y^{CFA}.

- Simple partial derivative

Denoting these missing components as $\dot{I}_x^l(P)$ and $\dot{I}_y^l(P)$, $l \neq k$ when $P \in S^k$, we can write:

$$\dot{\mathbf{I}}_x(P) = \begin{cases} \left(\dot{I}_x^R(P), I_x^{CFA}(P), \dot{I}_x^B(P)\right)^T & \text{if } P \in S^R \cup S^B, \\ \left(I_x^{CFA}(P), \dot{I}_x^G(P), \dot{I}_x^B(P)\right)^T & \text{if } P \in S^{G,R}, \\ \left(\dot{I}_x^R(P), \dot{I}_x^G(P), I_x^{CFA}(P)\right)^T & \text{if } P \in S^{G,B}, \end{cases} \tag{14}$$

and $\dot{\mathbf{I}}_y$ in the same way. We compute a missing derivative component $\dot{I}_x^l$ (or $\dot{I}_y^l$) at each pixel P by interpolation of the derivative values I_x^l (or I_y^l) of the same component available in a neighbourhood of P. This neighbourhood depends on the CFA pixel subset S^k to which P belongs. The four neighbourhood configurations used for this interpolation are defined in Fig. 3, and the partial derivatives then write as:

$$\dot{\mathbf{I}}_x(P) = \begin{cases} \left(\frac{1}{2} \cdot \sum_{Q \in N^x(P)} I_x^R(Q),\ I_x^{CFA}(P),\ \frac{1}{2} \cdot \sum_{Q \in N^y(P)} I_x^B(Q)\right)^T & \text{if } P \in S^R, \\ \left(I_x^{CFA}(P),\ \frac{1}{4} \cdot \sum_{P \in N^+(P)} I_x^G(P),\ \frac{1}{4} \cdot \sum_{P \in N^\times(P)} I_x^B(P)\right)^T & \text{if } P \in S^{G,R}, \\ \left(\frac{1}{4} \cdot \sum_{Q \in N^\times(P)} I_x^R(Q),\ \frac{1}{4} \cdot \sum_{Q \in N^+(P)} I_x^G(Q),\ I_x^{CFA}(P)\right)^T & \text{if } P \in S^{G,B}, \\ \left(\frac{1}{2} \cdot \sum_{Q \in N^y(P)} I_x^R(Q),\ I_x^{CFA}(P),\ \frac{1}{2} \cdot \sum_{Q \in N^x(P)} I_x^B(Q)\right)^T & \text{if } P \in S^B, \end{cases} \tag{15}$$

and:

$$\dot{\mathbf{I}}_y(P) = \begin{cases} \left(\frac{1}{2} \cdot \sum_{Q \in N^y(P)} I_y^R(Q),\ I_y^{CFA}(P),\ \frac{1}{2} \cdot \sum_{Q \in N^x(P)} I_y^B(Q)\right)^T & \text{if } P \in S^R, \\ \left(\frac{1}{4} \cdot \sum_{Q \in N^\times(P)} I_y^R(Q),\ \frac{1}{4} \cdot \sum_{Q \in N^+(P)} I_y^G(Q),\ I_y^{CFA}(P)\right)^T & \text{if } P \in S^{G,R}, \\ \left(I_y^{CFA}(P),\ \frac{1}{4} \cdot \sum_{Q \in N^+(P)} I_y^G(Q),\ \frac{1}{4} \cdot \sum_{Q \in N^\times(P)} I_y^B(Q)\right)^T & \text{if } P \in S^{G,B}, \\ \left(\frac{1}{2} \cdot \sum_{Q \in N^x(P)} I_y^R(Q),\ I_y^{CFA}(P),\ \frac{1}{2} \cdot \sum_{Q \in N^y(P)} I_y^B(Q)\right)^T & \text{if } P \in S^B. \end{cases} \tag{16}$$

For instance (see Fig. 2(c)), the four diagonal neighbours of $P(1,1) \in S^{G,B}$ are used to compute $\dot{I}_x^R(P)$ while its four closest neighbours are used to compute $\dot{I}_x^G(P)$. The x-derivative components missing at $P(2,3) \in S^B$ are interpolated from two of its closest neighbours (i.e., vertical neighbours for $\dot{I}_x^R(P)$ and horizontal ones for $\dot{I}_x^B(P)$).

- Deriche partial derivative

As for the simple partial derivative, we need to estimate the two derivative components missing at each pixel P in I_x^{CFAD} and I_y^{CFAD}:

$$\dot{\mathbf{I}}_x(P) = \begin{cases} \left(I_x^{CFAD}(P), \dot{I}_x^G(P), \dot{I}_x^B(P)\right)^T & \text{if } P \in S^R, \\ \left(\dot{I}_x^R(P), I_x^{CFAD}(P), \dot{I}_x^B(P)\right)^T & \text{if } P \in S^G, \\ \left(\dot{I}_x^R(P), \dot{I}_x^G(P), I_x^{CFAD}(P)\right)^T & \text{if } P \in S^B, \end{cases} \tag{17}$$

and $\dot{\mathbf{I}}_y$ in the same way. The missing derivative component $\dot{I}_x^l$ (or $\dot{I}_y^l$) with ($l \neq k$) at each pixel P is given by:

If $\left(|\frac{1}{2}\cdot\sum_{Q\in N^x(P)} I_x^G(Q)| > |\frac{1}{2}\cdot\sum_{Q\in N^y(P)} I_x^G(Q)|\right)$ in the case where $P \in (S^R \cup S^B)$:

$$\dot{\mathbf{I}}_x(P) = \begin{cases} \left(I_x^{CFAD}(P),\ \frac{1}{2}\cdot\sum_{Q\in N^x(P)} I_x^G(Q),\ \frac{1}{4}\cdot\sum_{Q\in N^\times(P)} I_x^B(Q)\right)^T & \text{if } P \in S^R \\ \left(\frac{1}{2}\cdot\sum_{Q\in N^x(P)} I_x^R(Q),\ I_x^{CFAD}(P),\ \frac{1}{2}\cdot\sum_{Q\in N^y(P)} I_x^B(Q)\right)^T & \text{if } P \in S^{G,R}, \\ \left(\frac{1}{2}\cdot\sum_{Q\in N^y(P)} I_x^R(Q),\ I_x^{CFAD}(P),\ \frac{1}{2}\cdot\sum_{Q\in N^x(P)} I_x^B(Q)\right)^T & \text{if } P \in S^{G,B}, \\ \left(\frac{1}{4}\cdot\sum_{Q\in N^\times(P)} I_x^R(Q),\ \frac{1}{2}\cdot\sum_{Q\in N^x(P)} I_x^G(Q),\ I_x^{CFAD}(P)\right)^T & \text{if } P \in S^B, \end{cases} \tag{18}$$

If $\left(|\frac{1}{2}\cdot\sum_{Q\in N^x(P)} I_x^G(Q)| < |\frac{1}{2}\cdot\sum_{Q\in N^y(P)} I_x^G(Q)|\right)$, $\dot{I}_x^G(P) = \frac{1}{2}\cdot\sum_{Q\in N^y(P)} I_x^G(Q)$ where $P \in (S^R \cup S^B)$.

and $\dot{\mathbf{I}}_y(P)$ is calculated in the same way as $\dot{\mathbf{I}}_x(P)$.

III.4. Canny edge detection

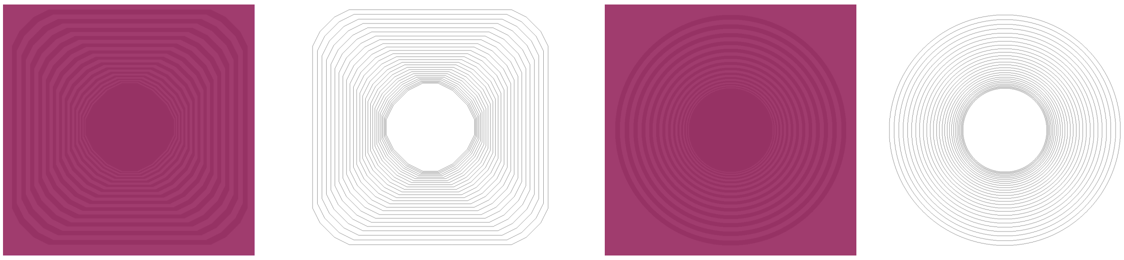
From the above fully-defined partial derivatives according to x and y, Di Zenzo's approach (see Sect. III.1) allows us to estimate the norm $||\nabla\mathbf{I}||$ and the direction θ^* of the vector gradient directly from CFA image. Computing binary edge image requires two steps: the extraction of the local maxima by removing the non local maxima of $||\nabla\mathbf{I}||$ along the direction θ^*, followed by local maxima image thresholding. We'll see more on this in part section IV.1.

IV. Results

This section demonstrates the relevance of using the CFA image in edge detection. This new approach of edge detection is compared with edge detection at demosaicing image.To evaluate the edge detection results, we have conducted a number of tests.

IV.1. Experimental procedure

To assess the robustness of different edge detection approach against edge orientation, we build a first synthesis image that contains several hexadecagons. We choose this image because it contains 16 different orientations belonging to $[-\pi, \pi[$ with a step of $\frac{\pi}{8}$, which coincides with the possible values that can take θ^* (See the section III.1). To take a larger range of edge orientation, we build a second synthesis image that contains several concentric circles (see fig 4). The two synthesis images contains 13 concentric circles (or hexadecagons) build around a main circle (or hexadecagon). The distance between the main circle (or hexadecagon) and the first concentric circle is setting by a pixel, this distance increases by one pixel for each new concentric circle (or hexadecagon). The distance between the last concentric circles (or hexadecagon) and the penultimate circle (or hexadecagon) will be 14 pixels. In the same way, the thickness of the first circle (or hexadecagon) is set to 1 pixel to reach 14 pixels for the last concentric circles (or



(a) hexadecagon (b) True edge (c) circles (d) True edge

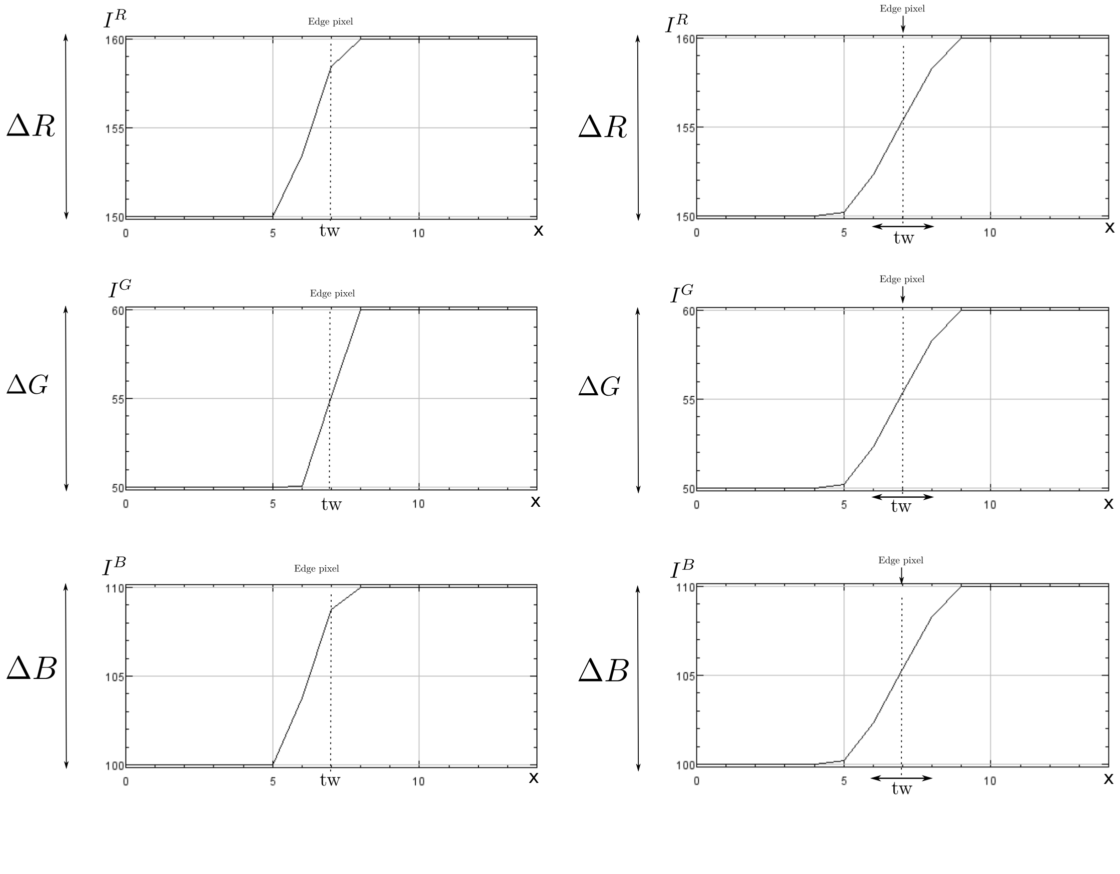
Figure 4: *Images of circles and hexadecagon.*

hexadecagon).

a Ground truth image

To build these reference images, we took into account several parameters which influence the edge detection, such as changes in colour component levels and transition width. The transition width depends of the standard deviation of a Gaussian Blur applied to the image (See fig 5). In the horizontal (or vertical) direction the standard deviation of the Gaussian $\rho = 0$ corresponds to a transition width $tw = 1$ pixel, standard deviation of the Gaussian $\rho = 1$ it corresponds to $tw = 3$ pixels and standard deviation of the Gaussian $\rho = 2$ it corresponds to $tw = 7$ pixels.

To measure the noise detection robustness, we corrupt the image by a additive Gaussian noise. Zhang et *al*. propose to add a noise independently for each channel [40]. Effectively, to add a Gaussian noise with a standard deviation σ to a colour image $\mathbf{I}$, we added separately to each



(a) $\rho = 0$, $tw = 1$

(b) $\rho = 1$, $tw = 3$

Figure 5: *width reference image:* $R = 150$, $G = 50$, $B = 100$, $\Delta R = \Delta G = \Delta B = 10$ *and noiseless.*

image channel I^k a Gaussian noise $\mathcal{N}^k$ with a standard deviation σ_k which is proportional to the energy E^k:

$$\sigma_k = \sigma \cdot \left(\frac{E^k}{E}\right) \quad \text{with } E^k \text{ Energy of the channel } k \tag{19}$$

Where: $E = E^R + E^G + E^B$

$k \in \{R, G, B\}$,

Therefore, the noisy channel I_N^k is obtained as: $I_N^k(x,y) = I^k(x,y) + \mathcal{N}^k(x,y)$.

b Measuring the edge detection quality

In literature, a widely used similarity measure between ground truth contours and detected edges is Pratt's Figure of Merit (FOM) [1]:

$$FOM = \frac{1}{max(card(I^T), card(I^E))} \times \sum_{P \in I^E} \frac{1}{1 + \frac{dis^2(P)}{9}} \tag{20}$$

where: I^T: true edge, I^E: detected edge and dis: the distance between the pixel P of I^T and the nearest pixel of I^E. The edge detection is considered as good if FOM is close to 1 and poor when this value tends to 0.

We saw in the section III.4 that the computing of the binary edge image requires the thresholding of the local maxima image. To choose the best threshold Th, we thresheld the local maxima image with all possible values of Th. Then we compute the Pratt's Figure of Merit criterion (FOM) between ground truth contours and each image obtained with the threshold Th. The selected threshold corresponds to the best FOM. So, for different edge detection approach, we have the best threshold corresponding to the best FOM. From the FOM criterion, we can evaluate the edge detection quality of different edge detection approach. We detail these different approaches and the evaluation results in the next section.

IV.2. Comparison and results

IV.3. Discusion

References

[1] I. Abdou and W. Pratt. Quantitative design and evaluation of enhancement/thresholding edge detectors. *Proceedings of the IEEE*, 67(5), May 1979.

[2] Hiroki Akiyama, Masayuki Tanaka, and Masatoshi Okutomi. Pseudo four-channel image denoising for noisy CFA raw data. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'15)*, pages 4778–4782, Québec City, Canada, September 2015.

[3] David Alleysson and Brice Chaix de Lavarène. Frequency selection demosaicking: A review and a look ahead. In *Procs. 20th IS&T/SPIE Electronic Imaging Annual Symposium (SPIE'08): Digital Photography IV*, volume 6822, San Jose, California, USA, January 2008.

[4] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679 – 698, November 1986.

[5] Lanlan Chang and Yap-Peng Tan. Hybrid color filter array demosaicking for effective artifact suppression. *Journal of Electronic Imaging*, 15(1):013003,1–17, January 2006.

[6] Chia-Hsiung Chen, Sao-Jie Chen, and Pei-Yung Hsiao. Edge detection on the Bayer pattern. In *Procs. IEEE Asia Pacific Conference on Circuits and Systems (APCCAS'06)*, pages 1132–1135, Singapore, December 2006.

[7] Xiangdong Chen, Liwen He, Gwanggil Jeon, and Jechang Jeong. Multidirectional weighted interpolation and refinement method for Bayer pattern CFA demosaicking. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(8):1271–1282, August 2015.

[8] Laurent Condat and Saleh Mosaddegh. Joint demosaicking and denoising by total variation minimization. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'12)*, pages 2781–2784, Orlando, Florida, USA, September 2012.

[9] Aram Danielyan, Markku Vehvilainen, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Cross-color BM3D filtering of noisy raw data. In *Proceedings of the 2009 International Workshop on Local and Non-Local Approximation in Image Processing (LNLA 2009)*, pages 125–129, Tuusula, Finland, August 2009.

[10] R. Deriche. Using canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, pages 167–187, 1987.

[11] Éric Dubois. Frequency-domain methods for demosaicking of Bayer-sampled color images. *IEEE Signal Processing Letters*, 12(12):847–850, December 2005.

[12] Joan Duran and Antoni Buades. Self-similarity and spectral correlation adaptive algorithm for color demosaicking. *IEEE Transactions on Image Processing*, 23(9):4031–4040, September 2014.

[13] Bart Goossens, Jan Aelterman, Hiêp Luong, Aleksandra Pižurica, and Wilfried Philips. Complex wavelet joint denoising and demosaicing using Gaussian scale mixtures. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'13)*, pages 445–448, Melbourne, Australia, September 2013.

[14] Bahadir K. Gunturk, Yucel Altunbasak, and Russell M. Mersereau. Color plane interpo-

lation using alternating projections. *IEEE Transactions on Image Processing*, 11(9):997–1013, September 2002.

[15] John F. Hamilton and James E. Adams. Adaptive color plan interpolation in single sensor color electronic camera. US patent 5,629,734, to Eastman Kodak Co., Patent and Trademark Office, Washington D.C., May 1997.

[16] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2013.

[17] Gwanggil Jeon and Éric Dubois. Demosaicking of noisy Bayer-sampled color images with least-squares luma-chroma demultiplexing and noise level estimation. *IEEE Transactions on Image Processing*, 22(1):146–156, January 2013.

[18] Daniel Khashabi, Sebastian Nowozin, and Jeremy Jancsary Andrew W. Fitzgibbon. Joint demosaicing and denoising via learned nonparametric random fields. *IEEE Transactions on Image Processing*, 23(12):4968–4981, December 2014.

[19] Daisuke Kiku, Yusuke Monno, Sunao Kikuchi, Masayuki Tanaka, and Masatoshi Okutomi. Residual interpolation for color image demosaicking. In *Proceedings of the IEEE International*

Conference on Image Processing (ICIP'13), pages 2304–2308, Melbourne, Australia, September 2013.

[20] Daisuke Kiku, Yusuke Monno, Masayuki Tanaka, and Masatoshi Okutomi. Minimized-Laplacian residual interpolation for color image demosaicking. In Nitin Sampat, Radka Tezaur, Sebastiano Battiato, and Boyd A. Fowler, editors, *Proceedings of the SPIE Electronic Imaging Annual Symposium (SPIE'14): Digital Photography X*, volume 9023, San Francisco, California, USA, February 2014.

[21] Daisuke Kiku, Yusuke Monno, Masayuki Tanaka, and Masatoshi Okutomi. Beyond color difference: Residual interpolation for color image demosaicking. *IEEE Transactions on Image Processing*, 25(3):1288–1300, March 2016.

[22] Jan Tore Korneliussen and Keigo Hirakawa. Camera processing with chromatic aberration. *IEEE Transactions on Image Processing*, 23(10):4539–4552, October 2014.

[23] Brian Leung, Gwanggil Jeon, and Éric Dubois. Least-squares luma–chroma demultiplexing algorithm for Bayer demosaicking. *IEEE Transactions on Image Processing*, 20(7):1885–1894, July 2011.

[24] Xin Li, Bahadir K. Gunturk, and Lei Zhang. Image demosaicing: a systematic survey. In

William A. Pearlman, John W. Woods, and Ligang Lu, editors, *Proceedings of the SPIE Conference on Visual Communications and Image Processing (VCIP'08)*, volume 6822, pages 68221J1–15, San Jose, California, USA, January 2008.

[25] Jingwei Liang, Jia Li, Zuowei Shen, and Xiaoqun Zhang. Wavelet frame based color image demosaicing. *Inverse Problems and Imaging*, 7(3):777–794, August 2013.

[26] Olivier Losson and Ludovic Macaire. Colour texture classification from colour filter array images using various colour spaces. *IET Image Processing*, 6(8):1192–1204, November 2012.

[27] Olivier Losson, Ludovic Macaire, and Yanqin Yang. Comparison of color demosaicing methods. *Advances in Imaging and Electron Physics*, 162:173–265, July 2010.

[28] Olivier Losson, Alice Porebski, Nicolas Vandenbroucke, and Ludovic Macaire. Color texture analysis using CFA chromatic co-occurrence matrices. *Computer Vision and Image Understanding*, 117(7):747–763, July 2013.

[29] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *Proceedings of the 12th International Conference on Computer Vision (ICCV'09)*, pages 2272–2279, Kyoto, Japan, September 2009.

[30] Daniele Menon and Giancarlo Calvagno. Color image demosaicking: An overview. *Journal of Signal Processing: Image Communication*, 26(8-9):518–533, 2011.

[31] Abdolreza Abdolhosseini Moghadam, Mohammad Aghagolzadeh, Mrityunjay Kumar, and Hayder Radha. Compressive framework for demosaicing of natural images. *IEEE Transactions on Image Processing*, 22(6):2356–2371, June 2013.

[32] Ibrahim Pekkucuksen and Yucel Altunbasak. Multiscale gradients-based color filter array interpolation. *IEEE Transactions on Image Processing*, 22(1), January 2013.

[33] Mattia Rossi and Giancarlo Calvagno. Luminance driven sparse representation based demosaicking. In *Proceedings of the IEEE International Conference on Image Processing (ICIP 2014)*, pages 1788–1792, Paris, France, October 2014.

[34] Tadashi Sakamoto, Chikako Nakanishi, and Tomohiro Hase. Software pixel interpolation for digital still cameras suitable for a 32-bit MCU. *IEEE Transactions on Consumer Electronics*, 44(4):1342–1352, November 1998.

[35] Tamara Seybold, Christian Keimel, Marion Knopp, and Walter Stechele. Towards an evaluation of denoising algorithms with respect to realistic camera noise. In *Proceedings of the IEEE*

International Symposium on Multimedia (ISM2013), pages 203–210, Anaheim, California, USA, December 2013.

[36] Lei Wang and Gwanggil Jeon. Bayer pattern CFA demosaicking based on multi-directional weighted interpolation and guided filter. *IEEE Signal Processing Letters*, 22(11):2083–2087, November 2015.

[37] Wei Ye and Kai-Kuang Ma. Color image demosaicing using iterative residual interpolation. *IEEE Transactions on Image Processing*, 24(12):5879–5891, December 2015.

[38] Silvano Di Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing*, 33(1):116–125, January 1986.

[39] Lei Zhang, Rastislav Lukac, Xiaolin Wu, and David Zhang. PCA-based spatially adaptive denoising of CFA images for single-sensor digital cameras. *IEEE Transactions on Image Processing*, 18(4):797–812, April 2009.

[40] Lei Zhang, Xiaolin Wu, and David Zhang. Color reproduction from noisy CFA data of single sensor digital cameras. *IEEE Transactions on Image Processing*, 16(9):2184–2197, September 2007.