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 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 only embed a single sensor. Such cameras are fitted with a Colour Filter Array (CFA) and their sensor delivers 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 of the literature are proposed by Gunturk et al. [15, 25] and most recently by Menon and Calvagno [31]. Since demosaicing is not our main concern and we precisely aim at avoiding it, we only give a quick insight of the main demosaicing strategies. 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 [28] 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 [27, 29]. 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.

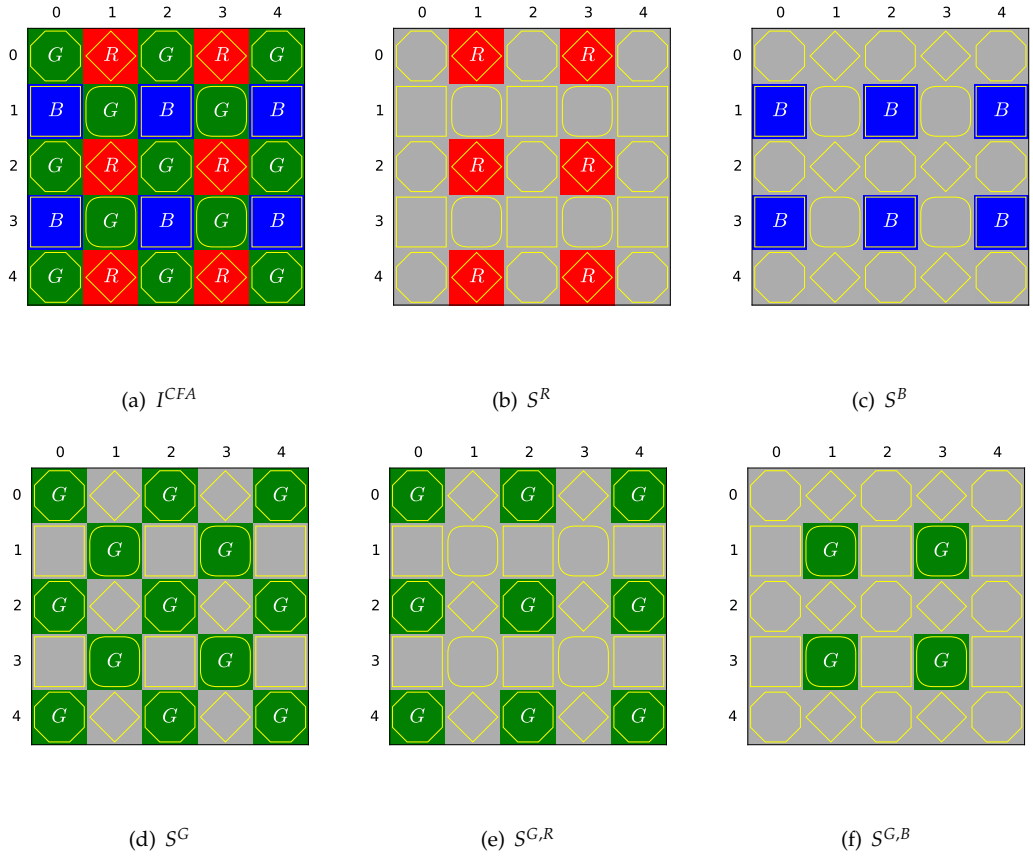
To perform edge detection directly on the CFA image, we adapt Di Zenzo's vector gradient [39] to the CFA image. The colour gradient is based on a tensor of the first derivatives of colour component images. Our main contribution is to adapt these derivative estimation to the CFA image where two colour component levels are missing at each pixel, as presented in section III. Indeed,

II. CFA demosaicing

II.1. Demosaicing problem formulation

Most of single-sensor colour 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

a quarter of all pixels arranged in a regular lattice. S^G contains half of all pixels arranged in a quincunx lattice, and 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 [35] is the simplest interpolation algorithm. It estimates each of the two colour components missing at a pixel by averaging the values of that component available in this pixel's neighbourhood. 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 [15], 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 is spatial correlation, that states to avoid using neighbour levels of different regions to interpolate missing components at a given pixel. 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 [31], from a fairly old patent by Hamilton and Adams [16] 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 [33] 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.

Still more recently have emerged another family of demosaicing schemes that rely on residual interpolation (RI) [20]. 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) [17] 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 [38]. Kiku et al. [20] replace the linear (horizontal or vertical) CD interpolation by an RI in the gradient-based threshold-free algorithm of Pekkucuksen and Altunbasak [33], and estimate colour planes

with each other as guides. For instance, to get $\hat{I}^R(P)$ at a pixel $P \in S^G$, 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 [38] iterate the RI process to refine $\hat{I}^G$. At each step, the horizontally- and vertically-estimated G channels are weighted by the inverse of the squared residual gradient. Kiku et al. [21, 22] 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 [37] propose a hybrid approach that first uses the CD-based weighted interpolation [7] over eight directions to estimate $\hat{I}^G$, then the GF with $\hat{I}^G$ as guidance and RI to estimate $\hat{I}^R$ and $\hat{I}^B$.

On the Kodak dataset that is widely used as a benchmark in the demosaicing literature, RI-based strategies perform slightly worse than state-of-the-art CD-based methods. But it is the opposite on the IMAX dataset, for which they even outperform methods that rely on compressive sensing and that require to learn a sparsifying dictionary [30, 32, 34]. This is because RI-based strategies less rely on spectral correlation and are less sensitive to sharp colour transitions that characterize IMAX images [13].

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 [12, 24]. Wavelet-based approaches (e.g., [26, 23]) and non-local image self-similarity approaches (e.g., [13]) 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 [40]. 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, 14, 19]. Although this approach improves the final image quality and provides relatively fast reconstructions, it prevents an inde-

pendent 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. [40] 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 similarities 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 first form four pseudo four-channel images, then transform each channel via PCA as in [40], and finally 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 [36], but the channel-dependent additive noise model (i.e., channel-wise signal-independent noises) used in [40] and [2] is a reasonable one for the white-balanced gamma-corrected raw signal [18].

III. Colour and CFA gradients

In this section, we first briefly recall how Di Zenzo [39] proposes 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 [39] 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 fundamen-

tal 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, the maximal variations of $\mathbf{I}$ are given by the closed-form solution $\theta^* = \frac{1}{2}\cdot\arctan\left(2b/(a-c)\right)$, $||\nabla\mathbf{I}||^2 = \frac{1}{2}\cdot\left(a+c+\sqrt{(a-c)^2+4b^2}\right)$.

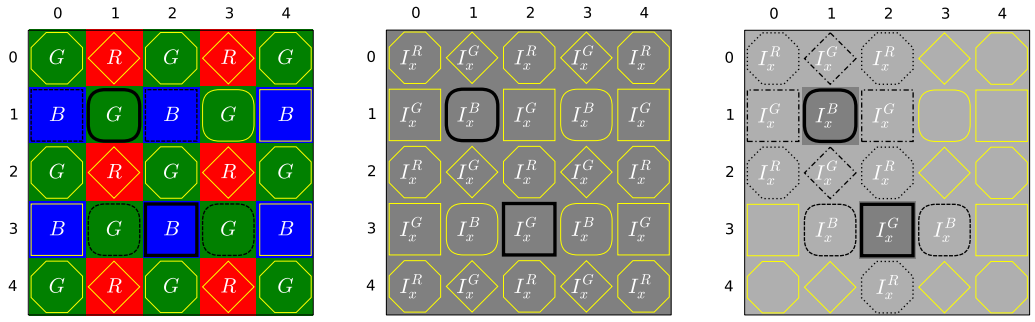
Below, we follow Di Zenzo's approach designed for a fully-defined (colour) image, and we adapt it to a CFA image. The key issue here is to estimate the partial derivatives of $\mathbf{I}$ from I^{CFA}, that we denote as $\dot{\mathbf{I}}_x$ and $\dot{\mathbf{I}}_y$. In the following two sections, we propose two ways to estimate these derivatives: either using a simple differentiation or Deriche's filter. In both cases, we first compute the partial derivatives I_x^{CFA} and I_y^{CFA} of I^{CFA} that is assumed continuous and derivable, which provides a single among the three derivative components of $\dot{\mathbf{I}}_x$ and $\dot{\mathbf{I}}_y$. The two missing ones are then estimated by interpolation.

III.2. Simple partial derivatives

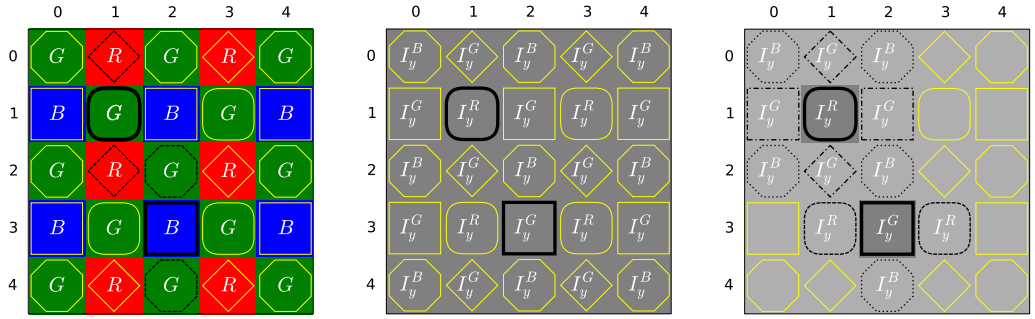
In this approach, we first compute the partial derivatives of I^{CFA} at each pixel $P(x,y)$ by simple differentiation:

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

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 $\mathbf{I}$ that would have been spectrally down-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 level difference matches with the horizontal (or vertical) simple 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

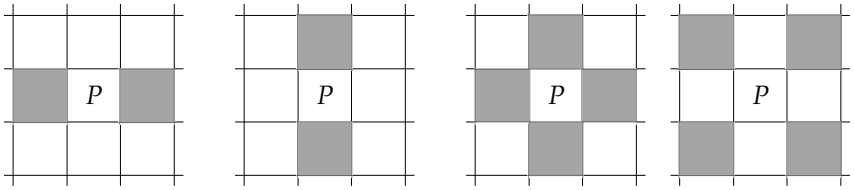


(a) Computation of I_x^{CFA} (b) I_x^{CFA} (c) Computation of i_x^l



(d) Computation of I_y^{CFA} (e) I_y^{CFA} (f) Computation of i_y^l

Figure 2: *First partial derivatives of a CFA image (see Eqs.* (4)– (7)*), 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.*

to:

$$I_x^{CFA}(P) = \begin{cases} I_x^G(P) & \text{if } P \in S^R \cup S^B, \\ I_x^R(P) & \text{if } P \in S^{G,R}, \\ I_x^B(P) & \text{if } P \in S^{G,B}, \end{cases} \quad \text{and} \quad I_y^{CFA}(P) = \begin{cases} I_y^G(P) & \text{if } P \in S^R \cup S^B, \\ I_y^B(P) & \text{if } P \in S^{G,R}, \\ I_y^R(P) & \text{if } P \in S^{G,B}. \end{cases} \tag{4}$$

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}. 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:

$$\dot{\mathbf{I}}_x(P) = \begin{cases} \left(\dot{I}_x^R(P), I_x^{CFA}(P), \dot{I}_x^B(P)\right)^T & \text{if } P \in S^R \cup S^B, \\ \left(I_x^{CFA}(P), \dot{I}_x^G(P), \dot{I}_x^B(P)\right)^T & \text{if } P \in S^{G,R}, \\ \left(\dot{I}_x^R(P), \dot{I}_x^G(P), I_x^{CFA}(P)\right)^T & \text{if } P \in S^{G,B}, \end{cases} \tag{5}$$

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 are then written as:

$$\dot{\mathbf{I}}_x(P) = \begin{cases} \left(\frac{1}{2}\cdot\sum_{Q\in N^x(P)} I_x^{CFA}(Q),\ I_x^{CFA}(P),\ \frac{1}{2}\cdot\sum_{Q\in N^y(P)} I_x^{CFA}(Q)\right)^T & \text{if } P\in S^R, \\ \left(I_x^{CFA}(P),\ \frac{1}{4}\cdot\sum_{P\in N^+(P)} I_x^{CFA}(P),\ \frac{1}{4}\cdot\sum_{P\in N^\times(P)} I_x^{CFA}(P)\right)^T & \text{if } P\in S^{G,R}, \\ \left(\frac{1}{4}\cdot\sum_{Q\in N^\times(P)} I_x^{CFA}(Q),\ \frac{1}{4}\cdot\sum_{Q\in N^+(P)} I_x^{CFA}(Q),\ I_x^{CFA}(P)\right)^T & \text{if } P\in S^{G,B}, \\ \left(\frac{1}{2}\cdot\sum_{Q\in N^y(P)} I_x^{CFA}(Q),\ I_x^{CFA}(P),\ \frac{1}{2}\cdot\sum_{Q\in N^x(P)} I_x^{CFA}(Q)\right)^T & \text{if } P\in S^B, \end{cases} \tag{6}$$

and:

$$\dot{\mathbf{I}}_y(P) = \begin{cases} \left(\frac{1}{2}\cdot\sum_{Q\in N^y(P)} I_y^{CFA}(Q),\ I_y^{CFA}(P),\ \frac{1}{2}\cdot\sum_{Q\in N^x(P)} I_y^{CFA}(Q)\right)^T & \text{if } P\in S^R, \\ \left(\frac{1}{4}\cdot\sum_{Q\in N^\times(P)} I_y^{CFA}(Q),\ \frac{1}{4}\cdot\sum_{Q\in N^+(P)} I_y^{CFA}(Q),\ I_y^{CFA}(P)\right)^T & \text{if } P\in S^{G,R}, \\ \left(I_y^{CFA}(P),\ \frac{1}{4}\cdot\sum_{Q\in N^+(P)} I_y^{CFA}(Q),\ \frac{1}{4}\cdot\sum_{Q\in N^\times(P)} I_y^{CFA}(Q)\right)^T & \text{if } P\in S^{G,B}, \\ \left(\frac{1}{2}\cdot\sum_{Q\in N^x(P)} I_y^{CFA}(Q),\ I_y^{CFA}(P),\ \frac{1}{2}\cdot\sum_{Q\in N^y(P)} I_y^{CFA}(Q)\right)^T & \text{if } P\in S^B. \end{cases} \tag{7}$$

For instance (see Fig. 2(c)), the levels of the four diagonal neighbours of $P(1,1) \in S^{G,B}$ are averaged to compute $\dot{I}_x^R(P)$ while those of its four closest neighbours are averaged 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 $\dot{I}_x^B(P)$).

III.3. Deriche's partial derivatives

We now propose a second way to estimate the first partial derivatives of $\mathbf{I}$ from I^{CFA} using Deriche's filters. In the literature, Deriche's derivation filters are widely used in Canny's gradient-

based edge detection approach because they are optimal according to three constraints: good detection, good localization, and low multiplicity of false detections [10]. The horizontal derivative results from a smoothing in the vertical direction and a derivation in the horizontal direction, while the vertical derivative uses the transposed directions. Moreover, each derivative operator may be effectively implemented by two recursive filters moving in opposite directions. To apply Deriche's filters on I^{CFA}, we are guided by this recursive implementation of the algorithm [11].

We first smooth the CFA image vertically as $\bar{I}_y^{CFA}(x,y) = \bar{I}_y^{CFA^-}(x,y) + \bar{I}_y^{CFA^+}(x,y)$, where the two terms result from a column-wise smoothing from top to bottom and from bottom to top:

$$\bar{I}_y^{CFA^-}(x,y) = a_1 \cdot I^{CFA}(x,y) + a_2 \cdot I^{CFA}(x,y-d) + b_1 \cdot \bar{I}_y^{CFA^-}(x,y-d) + b_2 \cdot \bar{I}_y^{CFA^-}(x,y-2d), \tag{8}$$

$$\bar{I}_y^{CFA^+}(x,y) = a_3 \cdot I^{CFA}(x,y+d) + a_4 \cdot I^{CFA}(x,y+2d) + b_1 \cdot \bar{I}_y^{CFA^+}(x,y+d) + b_2 \cdot \bar{I}_y^{CFA^+}(x,y+2d). \tag{9}$$

Working on the CFA image, we set $d = 2$, namely the spatial distance between the pixel $P(x,y)$ and its nearest neighbour that also belongs to S^k in the vertical direction (see Figs. 1(b)–1(d)). Note that missing levels of I^{CFA}, $\bar{I}^{CFA^-}$ and $\bar{I}^{CFA^+}$ are set to 0 in these equations, and that the weighting coefficients all depend on a single parameter α: $a_1 = \frac{(1-e^{-\alpha})^2}{1+2\cdot\alpha\cdot e^{-\alpha}-e^{-2\cdot\alpha}}$, $a_2 = a_1 \cdot (\alpha-1)\cdot e^{-\alpha}$, $a_3 = a_1 \cdot (\alpha+1)\cdot e^{-\alpha}$, $a_4 = -a_1 \cdot e^{-2\cdot\alpha}$, $b_1 = 2\cdot e^{-\alpha}$, $b_2 = -e^{-2\cdot\alpha}$.

Then, we compute the horizontal derivative of the CFA image as:

$$I_{x,D}^{CFA}(x,y) = -(1-\mathrm{e}^{-\alpha})^2 \cdot \left(I_{x,D}^{CFA^+}(x,y) + I_{x,D}^{CFA^-}(x,y) \right), \tag{10}$$

where the last two terms are obtained by scanning $\bar{I}_y^{CFA}$ from left to right and from right to left:

$$I_{x,D}^{CFA^-}(x,y) = \bar{I}_y^{CFA}(x-d,y) + b_1 \cdot I_{x,D}^{CFA^-}(x-d,y) + b_2 \cdot I_{x,D}^{CFA^-}(x-2d,y), \tag{11}$$

$$I_{x,D}^{CFA^+}(x,y) = -\bar{I}_y^{CFA}(x+d,y) + b_1 \cdot I_{x,D}^{CFA^+}(x+d,y) + b_2 \cdot I_{x,D}^{CFA^+}(x+2d,y). \tag{12}$$

At this stage, $I_{x,D}^{CFA}$ can be viewed as a down-sampled version of Deriche's horizontal derivative $\mathbf{I}_{x,D}$ of the reference image $\mathbf{I}$. In other words, $I_{x,D}^{CFA}(P)$ matches with $I_{x,D}^{k}(P)$ at each pixel $P \in S^k$. Like in the simple derivative case, we therefore need to estimate the two derivative components that miss in $I_{x,D}^{CFA}$ at each pixel P to get the fully-defined partial derivative $\dot{\mathbf{I}}_{x,D}$:

$$\dot{\mathbf{I}}_{x,D}(P) = \begin{cases} \left(I_{x,D}^{CFA}(P), \dot{I}_{x,D}^{G}(P), \dot{I}_{x,D}^{B}(P) \right)^T & \text{if } P \in S^R, \\ \left(\dot{I}_{x,D}^{R}(P), I_{x,D}^{CFA}(P), \dot{I}_{x,D}^{B}(P) \right)^T & \text{if } P \in S^G, \\ \left(\dot{I}_{x,D}^{R}(P), \dot{I}_{x,D}^{G}(P), I_{x,D}^{CFA}(P) \right)^T & \text{if } P \in S^B, \end{cases} \tag{13}$$

We compute these missing derivative components at each pixel P as:

$$\dot{\mathbf{I}}_{x,D}(P) = \begin{cases} \left(I_{x,D}^{CFA}(P),\ \frac{1}{2} \cdot \sum_{Q \in N(P)} I_{x,D}^{CFA}(Q),\ \frac{1}{4} \cdot \sum_{Q \in N^{\times}(P)} I_{x,D}^{CFA}(Q) \right)^T & \text{if } P \in S^R \\ \left(\frac{1}{2} \cdot \sum_{Q \in N^{x}(P)} I_{x,D}^{CFA}(Q),\ I_{x,D}^{CFA}(P),\ \frac{1}{2} \cdot \sum_{Q \in N^{y}(P)} I_{x,D}^{CFA}(Q) \right)^T & \text{if } P \in S^{G,R}, \\ \left(\frac{1}{2} \cdot \sum_{Q \in N^{y}(P)} I_{x,D}^{CFA}(Q),\ I_{x,D}^{CFA}(P),\ \frac{1}{2} \cdot \sum_{Q \in N^{x}(P)} I_{x,D}^{CFA}(Q) \right)^T & \text{if } P \in S^{G,B}, \\ \left(\frac{1}{4} \cdot \sum_{Q \in N^{\times}(P)} I_{x,D}^{CFA}(Q),\ \frac{1}{2} \cdot \sum_{Q \in N(P)} I_{x,D}^{CFA}(Q),\ I_{x,D}^{CFA}(P) \right)^T & \text{if } P \in S^B, \end{cases} \tag{14}$$

where $N(P) = N^x(P)$ if $|\frac{1}{2} \cdot \sum_{Q \in N^x(P)} I^{CFA}_{x,D}(Q)| > |\frac{1}{2} \cdot \sum_{Q \in N^y(P)} I^{CFA}_{x,D}(Q)|$ and $N(P) = N^y(P)$ otherwise.

The vertical derivatives $I^{CFA}_{y,D}(P)$ and $\dot{\mathbf{I}}_{y,D}(P)$ are computed in the same way as $I^{CFA}_{x,D}(P)$ and $\dot{\mathbf{I}}_{x,D}(P)$ but using transposed directions.

III.4. Conclusion

A reporter en section IV en compactant

In this section, we saw how the first partial derivatives are computed directly from CFA image. Firstly, a simple digital approximation is used to estimate these partial derivative. Then, we used the Deriche filter to compute the first partial derivatives from a CFA image. To assess the quality of edge detection obtained from CFA images, we need to compare it with that obtained from colour images. To estimate the simple or Deriche's partial derivatives of a colour image $\mathbf{I}$, we compute these partial derivatives in each colour channel I^k, $k \in \{R, G, B\}$. We compute the simple partial derivatives of I^k using simple digital approximations as in Eq. (3) and the Deriche's partial derivatives in the same way as for CFA images (see Sec. III.3) by setting d to 1. In fact, as the colour component image I^k is fully defined, each colour component is available at each pixel $P(x, y) \in I^k$. So, the spatial distance between the considered pixel and its nearest vertical

neighbour is set to $d = 1$ in Eqs. (8) and (9) for smoothing step, and in Eqs. (11) and (12) for derivative step. From the above 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 or from colour image.

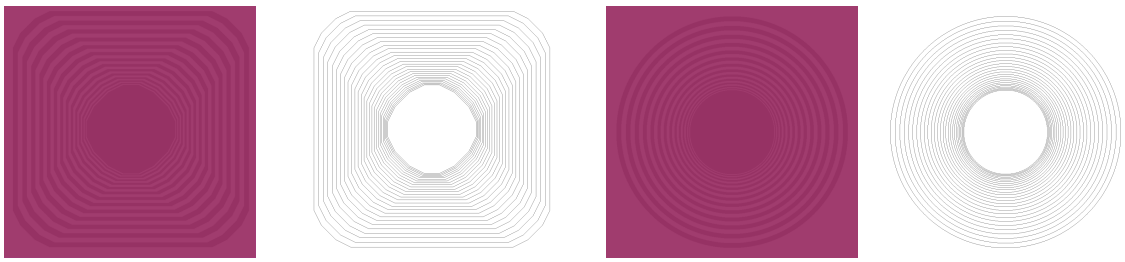
IV. Results

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 13 concentric hexadecagons (see fig.4(a)). We choose this image because it contains 16 different edge orientations belonging to $[-\pi, \pi[$ with a step of $\frac{\pi}{8}$ (see fig. 4(b)), that are the possible values of θ^* (See the section III.1). To take a finer step of edge orientation, we build a second synthesis image that contains 13 concentric circles (see fig 4(c)).

The two synthesis images contains 13 concentric shape (circle or hexadecagons) built around a main shape. To assess the quality edge detection and the quality edge localization, we vary the



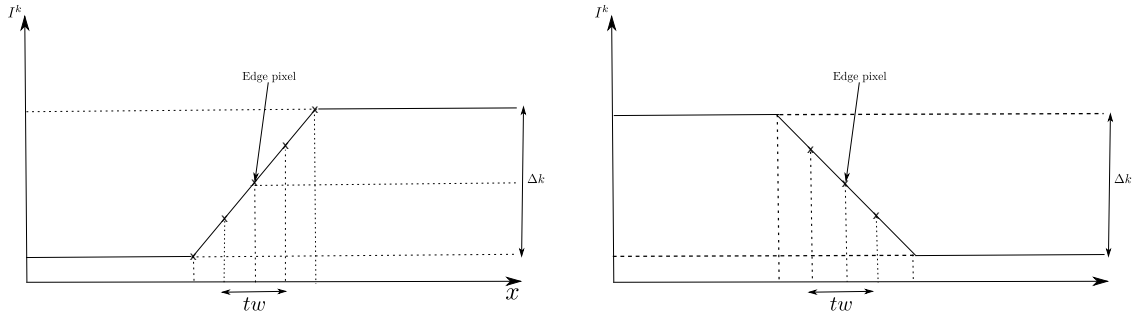
(a) hexadecagon (b) True edge (c) circles (d) True edge

Figure 4: *Images of circles and hexadecagon.*

diameter of the 13 concentric shape. Indeed, the diameter of the first concentric shape is equal to the diameter of the main shape more than a pixel, this diameter increases by one pixel for each new concentric shape (circle or hexadecagon). In the same way, the thickness of the first shape (circle or hexadecagon) is set to 1 pixel to reach 14 pixels for the last concentric shape. In fact, the first concentric shape where the distance between the concentric shape is very small (for instance 1 or 2 pixels), are used for assess the quality edge detection. The last concentric shape, where the distance between the concentric shape is larger, are used for assess the quality edge localization.

a Reference image

To build these reference images, we take into account several parameters which influence the edge detection, such as changes in colour component levels, the colour variation and transition width. We used two sets of colour component levels: ($R = 100$, $G = 150$, $B = 50$) and ($R = 150$, $G = 50$,



(a) with transition up

(b) with transition down

Figure 5: *width reference image.* $\rho = 1$, $tw = 3$ *and* $k \in \{R, G, B\}$

$B = 100$) because the added noise for each channel depends of the colour component levels. The compute of the partial derivative depends on the colour variation, for this reason we varied this colour variation that we denote Δk for $k \in \{R, G, B\}$. We took ΔG and $\Delta R \in \{10, 20, 30\}$ and $\Delta B = \Delta R$ because we assume that the variation is the same for the two chrominance components. 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 $\rho = 1$ corresponds to $tw = 3$ pixels and standard deviation of the Gaussian $\rho = 2$ corresponds to $tw = 7$ pixels.

To measure the noise detection robustness, we independently corrupt each channel I^k by additive Gaussian noise $\mathcal{N}^k$ [40]. Effectively, to add a Gaussian noise with a standard deviation

σ to a colour image $\mathbf{I}$, we added separately to each image channel I^k a Gaussian noise $\mathcal{N}^k$ with a standard deviation σ^k which is proportional to the energy E^k: $\sigma^k = \sigma \cdot \left(\frac{E^k}{E}\right)$ with E^k is the energy of the channel k (the mean value of I^k), σ ranging from 1 to 8 and $E = \frac{E^R+E^G+E^B}{3}$. Therefore, the noisy channel I_N^k: $I_N^k(x,y) = I^k(x,y) + \mathcal{N}^k(x,y)$.

b Edge detection quality assessment

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

$$FOM = \frac{1}{max(card(I^T), card(I^E))} \times \sum_{P \in I^E} \frac{1}{1 + \frac{dis^2(P)}{9}}, \tag{15}$$

where I^T is the true edge map of the reference image that we know precisely. I^E: detected edge obtained by thresholding the local maxima image(more details in section c). dis: the euclidean distance between each pixel P of I^E and its nearest pixel of I^T. The edge detection is considered as good when FOM is close to 1 and poor when this value tends to 0.

To take into account the sub- and the over-detection, we compute the mean FOM criterion of FOM^E that the Euclidean distance is computed between each pixel P of I^E and its nearest pixel of I^T and FOM^T that the Euclidean distance is computed between each pixel P of I^T and its nearest

pixel of I^E:

$$FOM = \frac{1}{2}(FOM^E + FOM^T), \tag{16}$$

where:

$$FOM^E = \frac{1}{max(card(I^T), card(I^E))} \times \sum_{P \in I^E} \frac{1}{1 + \frac{dis^2(P)}{9}}, \tag{17}$$

$$FOM^T = \frac{1}{max(card(I^T), card(I^E))} \times \sum_{P \in I^T} \frac{1}{1 + \frac{dis^2(P)}{9}}. \tag{18}$$

c 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. To choose the best threshold Th, we threshold the local maxima image with all possible values of Th. Then we compute the Pratt's Figure of Merit criterion (FOM) between ground truth 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

In this paper, we proposed two derivation methods on a CFA image, the simple derivative that we denote "SD" and Deriche's partial derivation. As seen in the previous section, the Deriche's partial derivation is calculated from two ways, using the spatial distance $d = 2$, we call this approach Sparse Canny-Deriche "SCD", and using the spatial distance $d = 1$ that we call Canny-deriche "CD". This last approach is only apply to the image fully defined (see the section III.3). To asses the quality edge detection in CFA image, we compared with the quality edge detection in colour images and luminance image. The colour images are obtained by the best demosaicing methods which are Pekkucuksen and Altunbasak [33], Kiku [20] et al and Zhang et al [**?**]. The luminance image is obtain with Dubois' approach [12] as explain in the section II.2. All these methods are tested in two cases, without denoising and with denoising proposed by Akiyama et al. [2] (see sectionII.3). For each edge detection approach, we looked for the best derivation method in terms of the FOM criterion as shown in the table below. Note that each cell of the table represents the mean FOM of the different reference images tested with the various parameters presented in section IV.1.

1. First, as shown in Tab.1, we are looking for the best derivation method for CFA images

mean FOM of CFA image		
	SD	CD
Without denoising (A)		
PFC denoising (B)		
$(A) \cup (B)$		

Ranking		
	SD	CD
Without denoising (A)		
PFC denoising (B)		
$(A) \cup (B)$		

Table 1: *The best partial derivative for CFA image*

2. Then, as shown in Tab.2, we are looking for the best derivation method for luminance images

mean FOM of luminance image			
	SD	SCD	CD
Without denoising (A)			
PFC denoising (B)			
$(A) \cup (B)$			

Ranking			
	SD	SCD	CD
Without denoising (A)			
PFC denoising (B)			
$(A) \cup (B)$			

Table 2: *The best partial derivative for luminance image*

3. For colour images, we must choose the best method of demosaicing coupled with the best derivation

method. The mean FOM obtained by each derivation method given in Table 3 are classified in Table 4

RGB image									
	Pekkucuksen			Kiku			Zhang		
Without denoising (A)									
PFC denoising (B)									
$(A) \cup (B)$									

Table 3: *The best partial derivative for luminance image*

RGB image									
	Pekkucuksen			Kiku			Zhang		
Without denoising (A)									
PFC denoising (B)									
$(A) \cup (B)$									

Table 4: *The best partial derivative for luminance image*

4. Once the best partial derivation method is chosen for each type of image, we compared the quality edge detection in CFA images with quality edge detection in colour image and luminance image. The results are given in table 5.

mean FOM			
	CFA-SCD	-CD	-CD
Without denoising (A)			
PFC denoising (B)			
$(A) \cup (B)$			

Classifying methods			
	CFA-SCD	-CD	-CD
Without denoising (A)			
PFC denoising (B)			
$(A) \cup (B)$			

Table 5: *Comparison of quality edge detection in CFA image with quality edge detection in colour and luminance images.*

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] R. Deriche. Fast algorithms for low-level vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1), January 1990.

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

[13] 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.

[14] 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.

[15] Bahadir K. Gunturk, Yucel Altunbasak, and Russell M. Mersereau. Color plane interpolation using alternating projections. *IEEE Transactions on Image Processing*, 11(9):997–1013, September 2002.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

[27] 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.

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

[29] 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.

[30] 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.

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

[32] 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.

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

[34] 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.

[35] 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.

[36] 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.

[37] 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.

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

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

[40] 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.

[41] 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.